

UNIVERSITÉ DU QUÉBEC À MONTRÉAL

AN EMPIRICAL INVESTIGATION OF TECHNOLOGICAL  
COLLABORATIONS IN SHARED R&D AND INNOVATION PROCESSES  
IN THE OPEN SOURCE SOFTWARE INDUSTRY

– TOWARDS AN INDUCTIVE THEORY OF OPEN AND COLLABORATIVE  
TECHNOLOGY PLATFORMS –

A THESIS SUBMITTED TO UNIVERSITÉ DU QUÉBEC À MONTRÉAL IN  
PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF  
DOCTOR OF PHILOSOPHY IN SCIENCE, TECHNOLOGY AND SOCIETY–STS

BY

SEYED AMIRHOSSEIN TAHERIZADEH

APRIL 2017

UNIVERSITÉ DU QUÉBEC À MONTRÉAL  
Service des bibliothèques

Avertissement

La diffusion de cette thèse se fait dans le respect des droits de son auteur, qui a signé le formulaire *Autorisation de reproduire et de diffuser un travail de recherche de cycles supérieurs* (SDU-522 – Rév.01-2006). Cette autorisation stipule que «conformément à l'article 11 du Règlement no 8 des études de cycles supérieurs, [l'auteur] concède à l'Université du Québec à Montréal une licence non exclusive d'utilisation et de publication de la totalité ou d'une partie importante de [son] travail de recherche pour des fins pédagogiques et non commerciales. Plus précisément, [l'auteur] autorise l'Université du Québec à Montréal à reproduire, diffuser, prêter, distribuer ou vendre des copies de [son] travail de recherche à des fins non commerciales sur quelque support que ce soit, y compris l'Internet. Cette licence et cette autorisation n'entraînent pas une renonciation de [la] part [de l'auteur] à [ses] droits moraux ni à [ses] droits de propriété intellectuelle. Sauf entente contraire, [l'auteur] conserve la liberté de diffuser et de commercialiser ou non ce travail dont [il] possède un exemplaire.»



UNIVERSITÉ DU QUÉBEC À MONTRÉAL

UNE ENQUÊTE EMPIRIQUE SUR LES COLLABORATIONS  
TECHNOLOGIQUES DANS LES PROCESSUS DE R & D ET D'INNOVATION  
DE L'INDUSTRIE DU LOGICIEL À CODE SOURCE OUVERT

- VERS UNE THÉORIE INDUCTIVE DES PLATESFORMES DE  
TECHNOLOGIE OUVERTE ET COLLABORATIVE -

THÈSE PRÉSENTÉE  
COMME EXIGENCE PARTIELLE  
DU DOCTORAT EN SCIENCE, TECHNOLOGIE ET SOCIÉTÉ (STS)

PAR  
SEYED AMIRHOSSEIN TAHERIZADEH

AVRIL 2017

## ACKNOWLEDGEMENTS

Many people encouraged, guided and supported me in my Ph.D. journey and particularly in the process of writing this dissertation. They influenced my work so profoundly, affected my personal development so strongly, and marked a lasting impression on me so solemnly that mere words cannot even begin to capture my deepest gratitude to them— for I am *forever* indebted to their positive impact on; and, selfless contribution to my life. Thus, in my simple words, I just mention names of those who have been with me in this journey and briefly describe their influence.

First and foremost, I would like to humbly thank my teacher, supervisor and mentor, emeritus Prof. Jorge Niosi, who has been always to me a source of creativity and novelty, positive encouragement, progress and continuity as well as pragmatism in academic research. He has been always an epitome of excellence in academic research and encouraged me to dream big, and act professionally.

Further, I am deeply grateful to Prof. Marc Banik, my teacher, my Mitacs industry research project director, and most of all, my mentor throughout the four years of my doctoral journey, and beyond. I forever remain indebted to him for his unwavering and steadfast commitment to make sure I realize my milestones in conformity with high academic standards, even throughout the difficult times when it was practically (and medically) not appropriate for him to involve. He transformed my theoretical research skills, industry approach, and academic professionalism in ways that I cannot even enumerate and indulge in describing them.

I also feel truly grateful to have the pleasure to be trained by Mr. Jérôme Oufella – Vice President of Technologies at Savoir-faire Linux. His genuine interest to teach me about the nuts and bolts of ICT sector, and in particular, open source software (OSS) technologies/industry coupled with his high level of expertise helped me build

a solid and invaluable understanding of OSS domain. Without his patience, dedication, and relentless support I would not have been able to deeply understand the nature of OSS development methodology, and the whole ecosystem around it. Indeed, I feel lucky to have been trained by a gentleman who is absolutely the best at what he does. Further, I extend my sincere thanks to the whole management team and software engineers at Savoir-faire Linux, for providing financial support, hosting me for a yearlong period, their heartfelt hospitality, kind support, and continuous collaboration to make sure I have been connected to all key knowledge sources needed.

On this note, I greatly appreciate the support provided by the administration/ leadership of OS communities (BigBlueButton, TikiWiki CMS Groupware, Odoo, Liferay, Ring Project, FFmpeg, Puppet, to name a few), foundations (e.g., Eclipse), and many individuals from both private and public sectors who have participated in this research. They sincerely and warmly worked closely with me during the data collection phases. They were more than cooperative and openly shared their time and expertise with me to make sure they have answered all my questions.

Next, I extend my sincere thanks to CIRST, Centre for Inter-University Research on Science & Technology, for providing intellectual support and partial financing throughout these four years. It was indeed a blessing to be embedded in this vibrant intellectual environment and network with the great minds in the field of science and technology.

Moreover, I am more than thankful to the individuals who played key roles in moving this research forward as well as helping me expand my personal and professional dimensions. I am grateful to Jean-Philippe Valois (Mitacs), Christophe Villemer and Cyrille Béraud (Savoir-faire Linux Inc.), Martine Foisy and Cloé Larivière (CIRST),

Joëlle Clément (Faculty of Humanities), Emmanuelle Vaast (McGill University), and many more.

Last but not least, I am grateful to Elahe and Yahya, my parents, for being a constant encouragement during this journey. In times of adversity, when I was stuck between a rock and a hard place, they kept reminding me of who I truly am, my core qualities, and values like perseverance, tenacity, integrity, and genuine desire to learn and move forward. I am also grateful to Sara, my wife, for feeling what I feel and respecting and supporting my choices. She carried the heavier load of parenting our twins, Emma and Hannah, so that I could complete my Ph.D. and this dissertation in less than four years.

I am also thankful to those individuals who helped me either directly or indirectly. I thank Samuel Sirois, a free software developer/consultant, for auditing my findings as an independent observer from industry and commenting on them. For the sake of brevity, I cannot mention all the names and affiliations that helped me accomplishing this project. However, beyond doubt, their supports and contributions have made their marks on my work and for that, I remain grateful.

## AVANT-PROPOS

This research has been an invaluable life experience both personally and professionally. On a professional level, while sharpening my already existing research skills such as data analysis, academic writing, etc., I had to carve several new precious dimensions into my professional character. For instance, as part of preparations for this research, I had to find a suitable industry partner, and justify why (on earth!) they had to invest both financially and intellectually in my research as well as collaborate with me in capacity of participant observer. In addition, I had to write a research proposal with both an industry focus as well as academic value-added and compete with a number of candidates from different universities and mainly from more research intensive fields such as chemistry, software engineering, mechanical engineering, electronics, etc. with more capacity for technological applications. Not to mention that as a doctoral candidate, a sophomore, it was indeed a daunting task to think both ways, especially under the stress of finding the financial silver lining to be able to finally complete my Ph.D. program. Thus, just the road to get the ball going expanded my overall capacity as an independent researcher, tested the limits of my perseverance, and tenacity, and engraved on me all I needed to start from the scratch my research program.

On a more personal note, it was a personal voyage of self-discovery, learning, unlearning, and relearning. Working in a new industry, and trying to develop an inductive theory challenged my learning and absorptive capacities for better. I developed the habit of being eager to venture into the unknown fields and making the effort to learn the new concepts. Even if, on the surface, they may have seemed unrelated to my immediate agendas, they may very well come in handy down the road. Not being able to see the connections immediately does not mean that there is none.

The triangulation process, for instance, helped me further realize and deeply appreciate the differences in individuals' perspectives and more importantly, why such differences existed in the first place. Thus, the process of learning about why OSS technological collaborations mean differently to different people and putting different pieces of the puzzle together, enlarged my learning capacity profoundly and equipped me with tools that I could use even in my personal life.

## DEDICATION

Dedicated to My Parents— Elahe and Yahya,

To My Twins— Emma and Hannah, the Universe and the Grace of My Life.

## TABLE OF CONTENTS

LIST OF FIGURES .....	xii
LIST OF TABLES .....	xiv
LIST OF ABBREVIATIONS .....	xvi
LIST OF THE RELATED RESEARCH PAPERS .....	xviii
ABSTRACT .....	xix
RÉSUMÉ .....	xxi
CHAPTER I	
INTRODUCTION .....	1
1.1 Introduction .....	1
1.2 Problem statement and study purpose .....	5
1.2.1 Research problem from academics' perspective .....	5
1.2.2 Research problem from a practitioners' perspective .....	10
1.3 The design of the research questions .....	15
1.4 Research methodology .....	18
1.5 The significance and relevance of the research .....	21
1.6 Limitations .....	24
1.7 Dissertation outline .....	25
CHAPTER II	
A CRITICAL REVIEW OF THE THEORETICAL PERSPECTIVES ON OPEN INNOVATION .....	28
2.1 Introduction .....	28
2.2 Open innovation concept and the changing landscape of innovation .....	28
2.3 'Openness': The essential condition to adopt open innovation paradigm .....	32
2.4 The theoretical perspectives on open innovation .....	38
2.4.1 Open innovation: Modes, models, and mechanisms .....	38
2.4.2 Overlap between user innovation and open innovation perspectives .....	43
2.4.3 Implications of private-collective action model for open innovation paradigm .....	48
2.5 Useful contributions of open innovation to open source software .....	53



2.5.1 What open innovation teaches us about open strategy .....	53
2.5.2 Open strategy .....	53
2.5.3 The relationship between BM and OIP .....	58
2.5.4 Business model: Concepts, definitions, and perspectives .....	60
2.5.5 Conventional BM concept poorly suited to explain distributed value creation .....	71
2.5.6 Does BM literature embrace OI? .....	71
2.5.7 Does OSS business models inform us about OBM concept? .....	77
2.6 To what extent OI is a valid theory to investigate the case of OSS .....	87
2.7 Conclusion .....	98
CHAPTER III	
PERSPECTIVES ON OPEN SOURCE SOFTWARE DEVELOPMENT .....	100
3.1 Introduction .....	100
3.2 The origin of OSS: The free/libre software movement and beyond .....	101
3.3 Significance of OSS as a case for open and collaborative technology development .....	108
3.4 The five inherent particularities of OSS technology .....	110
3.5 Theoretical perspectives on OSS development process as a technological innovation.....	112
3.5.1 Theories about why developers engage in OSS development process .....	113
3.5.2 The commercialization challenge of OSS.....	124
3.5.3 Understanding measures of ‘success’ in context of OSS.....	125
3.5.4 OSS innovation process .....	132
3.5.5 Organization of OSS projects .....	142
3.6 Conclusion .....	144
CHAPTER IV	
RESEARCH METHODOLOGY.....	146
4.1 Introduction.....	146
4.2 Methodological approach.....	146
4.2.1 Rationale for selecting the qualitative methodology and its three variations .....	152
4.2.2 Case study .....	153
4.2.3 Participant observation.....	154
4.2.4 Grounded theory approach.....	160

4.3 Research design.....	164
4.3.1 Defining the case and the unit of analysis.....	165
4.3.2 Sampling methods.....	168
4.3.3 Data collection methods: sources of evidence .....	169
4.3.4 Interview types.....	169
4.3.5 Research participants .....	173
4.3.6 Participant-observation .....	177
4.3.7 Background of participant-observation site .....	179
4.3.8 Documentation .....	182
4.3.9 Data analysis methods.....	183
4.3.10 Data analysis in the context of present study .....	190
4.4 Research evaluation .....	192
4.4.1 Validity .....	192
4.4.2 Generalizability or external validity .....	195
4.4.3 Reliability and ethical considerations .....	198
4.5 Methodological contribution.....	200
CHAPTER V	
RESULTS AND DISCUSSIONS: AN INDUCTIVE THEORY OF SUSTAINABLE OSS R&D AND INNOVATION PROCESS .....	202
5.1 Introduction.....	202
5.2 Concept of OSS technological collaborations .....	203
5.2.1 Background based on pilot interviews .....	203
5.2.2 Origins and properties of “Borromean Links” .....	204
5.2.3 OSSTC: Advantages, vulnerabilities and the relevance of the “Borromean Links” ....	208
5.2.4 Collaborative OSS R&D and innovation process .....	217
5.3 Constituents of OSS technological collaborations.....	220
5.3.1 Identifying actors and exploring the impacts of their diversity .....	222
5.3.2 Technological collaboration and its conceptual dimensions unpacked .....	253
5.3.3 Interdependence-Independence.....	295
5.3.4 Sustainability.....	301
5.3.5 Success.....	306

5.4 Putting categories in perspective: The core propositions and the theoretical model	311
5.4.1 Developing the core propositions .....	311
5.4.2 The theoretical model: Identifying the key factors influencing the success and sustainably of collaborative OSS RDIP .....	313
5.5 Discussions.....	315
5.6. Conclusion .....	319
CHAPTER VI	
FROM ALLIANCE FORMATION TO PLATFORM DEVELOPMENT: AN ALTERNATIVE PERSPECTIVE OF OSS TECHNOLOGICAL COLLABORATION.....	321
6.1 Introduction.....	321
6.2 OSS technological collaboration vis-à-vis strategic alliances concept.....	322
6.3 Overlaps between strategic alliances and OI: The shared conceptual caveats.....	325
6.4 Theories of strategic alliances and their implications for a theory of OSSTC.....	331
6.4.1 Transaction cost economics .....	331
6.4.2 Resource-based view .....	336
6.4.3 Resource dependence perspective.....	340
6.4.4 Knowledge-based theory of organizational capability.....	343
6.4.5 Relational view .....	347
6.5 Theoretical consolidation and possible explanations in the context of OSSTC ...	359
6.5.1 Actors, interdependence, and resources-based views .....	359
6.5.2 Technological collaboration and relational view .....	363
6.6 Open and collaborative technology platform as an alternative perspective.....	367
6.6.1 OSS as a case of open and collaborative technology platform– OCTP .....	369
6.6.2 Unit of analysis .....	372
6.6.3 Strategic resources and conduit for resources sharing .....	373
6.6.4 Governance .....	375
6.6.5 Collaboration logic.....	380
6.6.6 Learning .....	382
6.6.7 Primary sources of advantage .....	384
6.7 Conclusion .....	388

CHAPTER VII	
CONCLUSIONS AND RECOMMENDATIONS FOR FURTHER RESEARCH ....	389
7.1 Introduction.....	389
7.2 OSS technology: A useful empirical tool for developing OI and platform theories	389
7.3 Theoretical contributions: Bridging OSS, OI, OBM, and strategic alliances concepts.....	393
7.3.1 Platform perspective vs. Firm-centric-view .....	393
7.3.2 New knowledge co-development vs. Accessing and acquiring knowledge.....	394
7.3.3 A ‘detailed’ analysis of the nature of OSS value creation processes .....	395
7.3.4 Development of two detailed theories of OSSTC and their implications .....	398
7.3.5 Open and collaborative value creation platform as an example of OBM .....	400
7.4 Managerial implications.....	402
7.5 Limitations of the study .....	406
7.5.1 Research logistics and socio-cultural related issues.....	406
7.5.2 The researcher’s theoretical sensitivity: Merits and demerits .....	409
7.5.3 Challenges caused by abstraction of inductive theory building .....	411
7.6 Avenues for further research.....	412
APPENDIX .....	414
A SAMPLE OF THE INTERVIEW QUESTIONS.....	414
REFERENCES.....	417

## LIST OF FIGURES

Figure No.	Title	Page(s)
Figure 2.1	De-coupling the locus of innovation process	40
Figure 2.2	The open innovation model	42
Figure 2.3	Two forms of coupled open innovation	47
Figure 2.4	The six main components of a business model	66
Figure 2.5	The four major themes concerning BM studies	69
Figure 2.6	The BM mediates between the technical and economic domains	70
Figure 2.7	Conceptual framework of separation and overlap between OI, BM, and OBM concepts	75
Figure 4.1	Type of case study design including units of analysis	167
Figure 4.2	Main categories formed through inductive approach	192
Figure 4.3	An illustration of respondent validation: Excerpts from interview transcripts	194
Figure 5.1	Reduced diagrams of Borromean links in Tait's series	205
Figure 5.2	Some 3-component links from Tait (1876)	206
Figure 5.3	An illustration of Borromean links/Borromean rings	207
Figure 5.4	OSS development process: Interplay between evolution and inter-operability (compatibility)	213
Figure 5.5	A sample of build-root graph of software dependencies: "The dependency graph of the Ruby packages"	214
Figure 5.6	Conceptual interdependency between OSS technological collaboration and OSS collaborative R&D and innovation process	216
Figure 5.7	Water purifier model: Upstream-Downstream interaction and collaboration among developers	218
Figure 5.8	Three main actor categories of collaborative RDIP	223
Figure 5.9	Linear built-to-order collaborative R&D and innovation process	231

Figure 5.10	Relationship between clients' adoption of OSS solutions and the accrued benefits	240
Figure 5.11	OSS enterprise client typology	241
Figure 5.12	The vicious circle of conflicts leading to project forking	278
Figure 5.13	OSS technology development and diffusion typology and strategies	285
Figure 5.14	Seven stages of development from project proposal to proof of concept and beyond: The role and function of the Eclipse foundation	290
Figure 5.15	Institutional forms and institutional evolution of open source software	295
Figure 5.16	An Augmented View of Borromean Links Model of OSS Development	300
Figure 5.17	The theoretical model of OSS technological collaborations	314
Figure 6.1	The six functions of relationships embedded in OSSTC	363
Figure 6.2	A model of company-specific platform	371
Figure 6.3	A model of OSS OCTP	371

## LIST OF TABLES

Table No.	Title	Page(s)
Table 1.1	Research questions	17
Table 2.1	Contrasting principles of closed and open innovation	30
Table 2.2	Business model definitions	63-4
Table 2.3	The six main components of a business model	65
Table 3.1	Major motivations underlying contributions to OSS projects	122-3
Table 3.2	Basic statistics on major online community platforms	126
Table 4.1	List of participating firms	175-6
Table 4.2	List of participating OSS communities	177
Table 4.3	An illustration of open coding from field notes, observations, and interview transcriptions	186-7
Table 4.4	An illustration of open coding and categorizing	188
Table 5.1	List of the main categories accompanied by their subcategories	221
Table 5.2	Unpacking the concept of flexibility	235
Table 5.3	Unpacking of the concept of context for technological collaboration	255
Table 5.4	Conditions and consequences of the code sharing in OSSTC	262
Table 5.5	Typology of the relationships among firms and communities	264
Table 5.6	Breakdown of the concept of dual-purpose leadership	271
Table 5.7	An inductive theory of OSS technological collaborations: The core propositions	312
Table 5.8	An inductive theory of OSS technological collaborations: the core propositions	253

Table 6.1	The theories potentially relevant for explaining collaborative OSS R&D and innovation process	354-8
Table 6.2	An alternative perspective of OSSTC based on observations and emerging themes from OSS industry	378-9
Table 7.1	The summary of the key managerial implications	403-5



## LIST OF ABBREVIATIONS

ACRONYM	EXTENSION
BL	Borromean Links
BM	Business Model
BMI	Business Model Innovation
BT	Borromean Trinity
CCM	Constant Comparative Method
CMS	Content Management System
ERP	Enterprise Resource Planning
EU	European Union
FSF	Free Software Foundation
GTA	Grounded Theory Approach
IP	Intellectual Property
IRC	Internet Relay Chat
JV	Joint Venture
KBT	Knowledge-based Theory
NGO	Non-governmental Organization
OCTP	Open and Collaborative Technology Platform
OI	Open Innovation
OTP	Open Technology Platforms
OS	Open Source
OSS	Open Source Software
OSSBM	Open Source Software Business Model(s)
OSSD	Open Source Software Development

ACRONYM	EXTENSION
OSSTC	Open Source Software Technological Collaboration(s)
OIBM	Open Innovation-based Business Model
R&D	Research and Development
RBV	Resource-based View
RDP	Resource-Dependence Perspective
RV	Relational View
TCE	Transaction Cost Economics
UI	User Innovation
WPM	Water Purifier Model

## LIST OF THE RELATED RESEARCH PAPERS

The following refereed research papers have been produced during the course of writing this doctoral dissertation.

- Taherizadeh, A. (2016, July 6-8). Towards developing an eclectic and inductive theory of OSS technological collaborations as a *new breed* of strategic alliances. *The 16<sup>th</sup> International Joseph A. Schumpeter Society Conference on Evolutionary Economics*, Montreal, Canada.
- Taherizadeh, A. (2016, April 28-29). Swimming against the current: Why is 'open innovation' concept so rarely criticized? *The 4<sup>th</sup> International Conference on Innovation and Entrepreneurship (ICIE)*, Toronto, Canada. (ISSN: 2049-6834/ISBN: 978-1-910810-86-6)
- Taherizadeh, A., & Banik, M. (2014, June 8-11). Investigating coordination mechanisms of the Borromean Trinity of open source software development. *The XXV ISPIM Conference – Innovation for Sustainable Economy & Society* (ISBN: 978-952-265-590-5), Dublin, Ireland.
- Banik, M., & Taherizadeh, A. (2014, May 27-30). Investigating the Borromean Trinity of open source software innovation: Case study research for theory building on open innovation. *The Industry Studies Conference – Sustaining Sustainability in the Face of Increasing Risk*, Portland, U.S.A.

## ABSTRACT

Open source software (OSS) technologies have become increasingly important in the information and communication sectors, materialized for example, in the form of customized software solutions, but also community-developed web browsers and device systems such as Android, derived from open source Linux. Among scholars, OSS has also spurred a great deal of theoretical tension in management, organization science and economics. As an example, some of the emerging questions are those relating to the motivations of OSS developers, their incentives for benevolent contributions, the systems of governance put in place and how OSS can be the basis for a private business model. Such inquiries have led to the development of synthetic theories such as “private-collective innovation model” (von Hippel & von Krogh, 2003). Despite these efforts, many key questions relating to the mechanisms of collaboration of OSS firms remain unanswered by extant strategic management and economics literature. One of the reasons for this has do with the nature of the technology. OSS is a highly modular technology that shares many properties of a public good, in addition to being highly dependent on externalities (network and organizational learning). These conditions depart significantly from the basic assumptions underlying economic (e.g., transactions cost, resource based view) theories of management that have been used to explain strategic alliances and collaborations in other industries.

This research aims to explore, describe, and explain the multifaceted notion of OSS technological collaborations (OSSTC) of OSS R&D and innovation processes (RDIP). More specifically, it seeks to provide a finer-grained understanding of OSS value creation processes, how they are interrelated to one another, and in what ways their interdependencies influence the overall success and sustainability of collaborations.

The research employs interview and secondary data to uncover the process by which for-profit OSS firms collaborate – often without formal agreements – with OSS communities in order to develop OSS solutions for their clients. Using grounded theory (Strauss & Corbin, 1990; Glaser & Strauss, 1967) and participant observation techniques of data collection and analysis, this research unpacks the logic (e.g., so called “Borromean Links” and “water purifier” models) behind co-development and co-maintenance of OSS platforms. It also discusses how the context of such

collaborations affects the sustainability of OSS projects. Finally, it proposes the mechanisms by which technology management, coordination, and leadership functions are achieved when OSS firm project leaders occupy dual-purpose leadership roles, supporting both community and their private interests so as to best develop the latent value of the technology.

Results of the analysis of the qualitative data (e.g., axial and selective coding) are presented in terms of a gestalt model formed by five main conceptual categories. These categories correspond to the main factors (i.e., conceptual constructs) that influence success and sustainability of firm and community collaborations. Emerging from this analysis are theoretical propositions which are further refined in form of empirical hypotheses to be tested in further research. These propositions suggest, how the types of actors and the role they play within the community can impact the nature and sustainability of OSS firm-community collaborations, as well as the long-term sustainability of the resulting projects.

Finally, these findings are compared and contrasted to the strategic alliance literature, so as to better delineate and define them with respect to this received theory. One emerging difference is that while the strategic alliances literature views inter-partner learning as a race (among partners) which can negatively influence the cohesion and viability of a joint venture, the OSS collaborations case illustrates just the opposite. It suggests that inter-partner learning has a positive spillover effect which can strengthen the collaborative development of an OSS project having strong network effects (e.g., an infrastructure platform). Stated in terms of economic theory, the need for dynamic learning equilibria in OSS justifies continuous inter-partner learning made possible through firm-community collaborations. In this regard, this research makes contributions to the fields of open source software management, open innovation and open business model theory. It has also implications for managerial practice in OSS organizations and public agencies.

**Keywords:** Open source software innovation, open source management, inter-firm collaboration, technology platform development, strategic alliances, Linux, community development, software R&D, grounded theory methods.

## RÉSUMÉ

Les technologies à code source ouvert représentent maintenant une plus grande part du secteur des technologies de l'information et de la communication (TIC). Cette expansion se concrétise de différentes façons : solutions applicatives personnalisées, navigateurs Web développés par la communauté et même des systèmes d'exploitation pour appareil mobile tel qu'Android – lui-même une solution dérivée du noyau Linux. À l'intérieur même de la communauté universitaire, les logiciels à code source ouvert ont également bouleversé beaucoup de théories dans des domaines aussi divers que la gestion, l'organisation scientifique et la science économique. À titre d'exemple, pensons à certaines questions émergentes vis-à-vis les différents éléments de motivation des développeurs de logiciels à code ouvert, ce qui les incite à faire des contributions sans rémunération monétaire directe (bénévolement), les systèmes de gouvernances mis en place pour une saine gestion de projets ou encore comment des entreprises privées peuvent utiliser les technologies à code source ouvert comme base d'un modèle d'affaires rentable. De tels questionnements ont mené au développement de théories comme « private-collective innovation model » (von Hippel et von Krogh, 2003). Malgré ces efforts, la littérature économique ainsi que la littérature sur les stratégies de gestion laissent de nombreuses questions clefs relatives aux mécanismes de collaboration d'entreprises privées ayant les technologies à code source ouvert comme modèle d'affaires sans réponse. Une des raisons à cette absence de réponse est due à la nature même des technologies. Les technologies à code source ouvert sont des technologies très modulaires qui partagent beaucoup de caractéristiques avec le bien commun en plus d'être fortement couplés avec des dépendances externes telles que le réseau et l'apprentissage organisationnel. Ces deux caractéristiques à elles seules démarquent significativement les technologies à code source ouvert des hypothèses de bases proposées par les théories de la gestion économique. (ex. : le coût des échanges et la gestion basée sur les ressources) ainsi que les hypothèses proposées par les théories de la gestion utilisées pour expliquer les alliances stratégiques et les collaborations mises en place dans d'autres industries.

Cette recherche se donne comme objectif d'explorer, de décrire et d'expliquer les différentes facettes de la collaboration telle que proposée et appliquée dans le domaine des technologies à code source ouvert ainsi que les processus de recherche et développement et d'innovation. Plus précisément, cette recherche veut fournir une compréhension plus précise des processus de création de valeur par les technologies à

code source ouvert, de la manière dont ils sont interdépendants et de quelle manière leurs interdépendances influence le succès général et la durabilité des collaborations.

Cette recherche utilise un processus d'entrevues et des données secondaires pour faire émerger le processus par lequel les entreprises à but lucratif en technologie à code source ouvert collaborent – souvent sans accords formels – avec la communauté afin de développer des solutions à code source ouvert pour leurs clients. En utilisant la théorie ancrée (Strauss et Corbin, 1990, Glaser et Strauss, 1967) et des techniques d'observation participative pour collecter et analyser les données, cette recherche met en lumière et explique la logique collaborative (ex. : le modèle des « anneaux borroméens » et le modèle du « purificateur d'eau ») derrière le codéveloppement et le soutien collaboratif des plate-formes à code source ouvert. Elle explique également comment le contexte de ces collaborations affecte la durabilité des projets à code source ouvert. Enfin, cette recherche propose un modèle par lequel les fonctions de gestion, de coordination et de leadership sont atteintes lorsqu'une entreprise joue un rôle de leader en soutenant à la fois les intérêts de la communauté et ses propres intérêts privés afin de développer au mieux la valeur technologique latente.

Les résultats de l'analyse des données qualitatives (ex. : par codage axial et codage sélectif) sont représentés par des modèles gestaltistes formés principalement par cinq catégories conceptuelles. À partir de cette analyse, il a été possible de dériver différentes propositions théoriques qui ont été raffinées sous forme d'hypothèses empiriques à être testées dans le cadre de recherches plus approfondies. Ces propositions définissent comment les différents acteurs et le rôle qu'ils jouent à l'intérieur de la communauté influencent substantiellement la nature et la durabilité des collaborations entre les entreprises privées et la communauté des technologies à code source ouvert ainsi que la pérennité du succès des projets qui en découlent.

Finalement, ces résultats sont comparés et confrontés aux théories proposées par la littérature sur les alliances stratégiques afin de les décrire de façon plus robuste et les définir en fonction des théories acceptées. Une différence notable est que la littérature sur les alliances stratégiques considère l'apprentissage inter-partenaires comme une course (entre lesdits partenaires) qui peut influencer négativement la cohésion et la viabilité d'une alliance stratégique alors que les cas de collaboration dans le domaine des technologies à code source ouvert démontrent le contraire. Cette dernière suggère plutôt que l'apprentissage inter-partenaires a des retombées positives qui peuvent renforcer la collaboration dans le cadre du développement d'un projet à code source

ouvert. Conformément à la théorie économique, la nécessité d'équilibrer les apprentissages dynamiques dans le cadre de projet à code source ouvert justifie un apprentissage continu entre les partenaires rendu possible par des collaborations solides entre les entreprises privées et la communauté. À cet égard, cette recherche apporte des contributions aux domaines de la gestion de projets à code source ouvert, de l'innovation ouverte et de la théorie des modèles commerciaux ouverts. Cette recherche apporte également une contribution sur l'application pragmatique de modèle de gestion dans les organisations basées sur un modèle ouvert ainsi que dans les organismes publics.

Mots clefs : innovation en logiciel à code source ouvert, gestion de projet à code source ouvert, collaboration inter-entreprises, développement de plate-formes technologies, alliances stratégiques, GNU/Linux, communauté, programmation, recherche et développement logiciel, méthode de la théorie ancrée



## CHAPTER I

### INTRODUCTION

#### 1.1 Introduction

Open Source Software (OSS) is software technology created collectively through communities of developers who collaborate to develop the software that they or their organizations need. It has been a major cultural, economic and social phenomenon (e.g., von Hippel & von Krogh, 2003), and it has spurred theoretical tension as it deviates *sharply* from the predictions and explanations of existing theories in different fields of economics, sociology, and organization science (e.g., von Krogh & Spaeth, 2007; Bonaccorsi & Rossi, 2003), and software development economics (Ajila & Wu, 2007). Its social impact is measured by turning software into a public good (Holtgrewe & Werle, 2001), while its economic impact has been that of revolutionizing the Information and Communications Technologies (ICT) sector and all other downstream industries that feed on it (Fosfuri, Giarratana, & Luzzi, 2008). OSS reduces the social loss resulting from restrictions posed on software modification by allowing users to freely learn, and by offering a novel model for creating and capturing value in ICT. As a testament to these benefits, consider that a majority of the European governments have already shifted from deploying proprietary software to OSS, in order to gain autonomy, control, as well as reduce costs and enhance their ICT's privacy and security. Consequently, OSS has become a preferred option due to offering security, reliability, flexibility, stability, control, auditability and quality at lower cost, especially in the last decade in which we have been observing the emergence of *strategic* and *commercial* OSS (Fitzgerald, 2006).

A few examples shall clarify this issue further. For instance, in developing countries they use OSS to prolong the life of computers in schools when costs of upgrading required by proprietary providers are simply unbearable. Similarly, the Ville de Montréal was faced with dilemma of spending about 7 M\$ on migration from Windows XP to the Windows 7; or, to switch to OSS which is royalty free (Normandin, 2014). Likewise, to ensure flight safety, airliners have integrated OSS modules into their ICT systems so much so that from cockpit control system to seat monitors, all devices (e.g., a range of embedded systems) run on OSS. The logic is actually simple: once you have unrestricted access to every line of code, you have freedom to modify it, and guarantee security of the overall system. This immeasurable impact of OSS is the result of applying 'open methodology' to software research and development (R&D) and innovation process (RDIP). This methodology also spurs and facilitates the collective effort of the network of individuals to innovate and it leads to a faster diffusion of innovations.

However, while the transparency as the fruit of openness can create opportunities for improvements and further technological developments, the actual implementation of OSS development methodology can at times be problematic. If the heterogeneous skills, incentives and expectations of contributing individuals, private as well as public organizations are not effectively coordinated and led towards achieving milestones, success of OSS Technological Collaborations (OSSTC) and sustainability of the collaborative R&D efforts can be nothing but a mirage. Therefore, the Achilles' heel of OSS projects is not only the lack of resources, but also aligning incentives, coordinating as well as effectively leading resources so as to enable OSS projects to gain attraction in the market where an OSS product may compete not only with a proprietary (i.e., mainstream) one, but also with other OSS products.

Past research has looked into the emergence and survival of the OSS phenomenon from different angles. Several authors have focused their attention on the

'*motivations*' which induce behavior in users and developers and explain the underlying reasons for their contributions to OSS projects (Lerner & Tirole, 2002; Lakhani & Wolf, 2005; Roberts, Hann, & Slaughter, 2006). Others have been intrigued by '*governance*' issues within OSS communities (Kogut & Metiu, 2001; Shah, 2006; O'Mahony & Ferraro, 2007; Schaarschmidt, Walsh, & von Kortzfleisch, 2015). Some other researchers have found the '*organization*' of OSS projects and communities of importance and devoted their efforts to explore the topic (Scacchi 2002; West & O'Mahony, 2005). Yet a few researchers (e.g., Feller & Fitzgerald, 2000; von Krogh et al., 2003; von Hippel, 2005) have made an effort to explore and unbox the unorthodox *innovation process* of OSS development.

Despite these insightful efforts, we still know little about this complex and multifaceted form of collaborative technology development (see, Linux Foundation, 2014). As a case in point, in the seminal work of von Hippel and von Krogh (2003), authors try to resolve the perplexing issue of OSS innovation model by arguing that OSS represents a middle ground solution. It is a happy marriage between private investment model (Demsetz, 1967) and collective action model (Olson, 1967). This third option is called "private-collective" innovation model that offers enough incentives for both models to coexist. However, the proposed hybrid model invites further research concerning the *role and nature of leadership* in relation to sustaining the ongoing activity in OSS projects and communities. For example, some authors believe that as the OSS community gets larger and larger, it becomes more obvious that they need to establish a central authority or leadership in order to monitor the members' activities and impose sanctions on free-riders (Hardin, 1982; Swanson, 1992). The importance of leadership role in developing OSS projects has been already raised (Pavlicek, 2000); yet, the nature of leadership seems to have remained

as an elusive concept<sup>1</sup>. For example, one of the norms of hacker community is that the to-be-done tasks cannot be forced by the leader (Raymond, 1999; Himanen, Torvalds, & Castells, 2001; Kelty, 2008); and the intricacies as how the job is done have not been illuminated.

Furthermore, we have fuzzy knowledge and limited understanding about several key OSS-related issues such as the following. For instance, the nature of technological or commercial success and the factors contributing to it is still underexplored. Similarly, we still lack clear understanding about what makes OSS R&D and innovation process (RDIP) a sustainable process, especially when sustainability is central to avoid having orphan OSS projects. In a similar vein, despite hearing that OSS offers ‘flexibility’ and ‘cost reduction’ benefits through different media and publications (see, e.g., Linux Foundation, 2014), we still do not know if these mean the same to different enterprise or individual users. Nor do we know what factors significantly contribute to the formation of these constructs, ‘how’; and, ‘under which conditions’ firms, organizations and individual can fully benefit from them. For example, if a firm lacks the necessary technological capability to connect and collaborate with OSS platform participants, then they may not be able to optimally benefit from ‘flexibility’ as enabled by the openness associated with OSS technology platforms. On the other hand, if creating such capability is not economically feasible or well justified, then flexibility is not an advantage any more. Above all, we need to have a theoretical frame of reference that comprehensively connects the dots, and meaningfully relates and binds together all vital interdependent elements at micro level so that we make sense of the whole R&D and innovation system.

Thus, in spite of the significance and ubiquity of OSSTC and increasing academic and commercial interest in its adoption by firms, governments and individuals (see,

---

<sup>1</sup> Some exceptions to this argument include, for example, the paper published by Li, Tan, and Teo (2012) who investigate the relationship between an OSS project leader’s leadership style and a developer’s motivation to contribute to the software development.

e.g., Linux Foundation, 2014, 2016), we have still inadequate detailed understanding of the subject matter. In line with this argument, this research aims at *exploring*, *describing*, and *explaining* the multifaceted notion of OSSTC embedded in OSS RDIP through adopting a mix of qualitative methodological approaches.

## 1.2 Problem statement and study purpose

### 1.2.1 Research problem from academics' perspective

#### A) Is OI paradigm an adequate theory?

Authors have tried to understand and explain OSS technology development process from different perspectives: social capital perspective (Méndez-Durón & García, 2009), motivational issues (Lerner & Tirole, 2002), project governance (Kogut & Metiu, 2001; O'Mahony, 2003), organizational issues (West & O'Mahony, 2005). However, more recently, the literature of distributed and collaborative R&D and innovation (e.g., von Hippel & von Krogh, 2003), has tightly associated OSS with the concept of OI (e.g., Gruber & Henkel, 2006; Maxwell, 2006; Chesbrough & Appleyard, 2007).

This association has extended to the point that several authors perceive OSS as “the poster child” or “a great exemplar” of OI concept, and treat OSS as a fertile ground to study and theorize about OI (Chesbrough & Appleyard, 2007; West & Gallagher, 2006; Morgen & Finnegan, 2010; Lundell & van der Linden, 2013). Even in the conceptual intersection of OI and OI-based BM or open business model (OBM), OSS has been popularized through the lens of OI. It has been studied as a showcase by those whose objective is to look deep into the OSS firms' processes and explain how OBM functions (e.g., Perr et al., 2010). More specifically, OI has become a more

relevant framework to study how private enterprises have ventured into exploiting the opportunities offered by OSS sector (West & Gallagher, 2006).

Although the past research attempts<sup>2</sup> have been insightful, to a certain degree, with regards to clarifying the concept of OI (and largely drawing on empirical data from OSS projects), there are still caveats associated with the past research which cast doubts on the validity and relevance of some arguments and provide impetus for further research on the overlap or connection between OSS and OI.

Fundamentally, open innovation paradigm (OIP) is criticized on three major fronts. It lacks *solid theoretical underpinning* (e.g., Lichtenthaler, 2011; van de Vrande et al., 2009) – an issue which is still valid (Vanhaverbeke & Cloudt, 2014). OI has been built upon fuzzy pillars so much so that it has been labelled for being “*old wine in new bottles*” (Trott & Hartmann, 2009) – meaning that, such thing as closed innovation (which has been used to justify OI) has never existed. Moreover, there are several research shortcomings, many uncharted research domains, and dearth of external validity of the concept.

Based on these caveats, one should take the usefulness and appropriateness of OI concept in studying OSSTC with a grain of salt. This has, therefore, motivated me to further examine the OI concept, its relevance and its usefulness in relation to OSSTC. For instance, although OSS projects (or their development technological platforms) demonstrate a great deal of openness – a characteristic shared with OIP– they do not resemble conventional enterprises or organizations. Therefore, a firm-based or dominantly firm-centric approach such as OI (see, Piller & West, 2014, p. 29) may not be able to explain the underlying logic and associated complexities of OSSTC and their supporting platforms.

---

<sup>2</sup> For example: Gassmann, (2006); Henkel (2006); West and Gallagher (2006); Dittrich and Duysters (2007); van de Vrande et al. (2009); Enkel et al. (2009); Gassmann et al. (2010).

B) Do conventional strategic alliances adequately explain OSSTC?

An OSSTC enables participants to: a) find optimum solutions to complex programming issues; b) reduce cost and duration of software R&D; c) mitigate the risk associated with software development and allay uncertainty with new software inventions; and d) learn know-how and capabilities which cannot be easily codified and learned otherwise. Therefore, when the underlying rationales for collaboration are acknowledged, we may observe that OSS collaborations share some resemblance with 'strategic/technological alliances/collaborations'<sup>3</sup>, and that they can be explained by the similar theories (e.g., transaction cost economics or resource-based view) that have already illuminated strategic alliances.

However, the literature on OSS has shed light on informal, non-contractual, and trust-based/social relationships of collaborators as conduits for collaboration, knowledge sharing and knowledge creation (e.g., Lerner & Tirole, 2002; ; Bonaccorsi & Rossi, 2003; von Hippel & von Krogh, 2003; Mendez-Duron & Garcia, 2009). Such distinguishing characteristics are basis for a *theoretical dilemma* that does not let us sharply categorize OSSTC under strategic alliances that are mainly studied as R&D elements governed by long-term legally binding contracts (e.g., Niosi, 1995).

Thus, in order to better understand and resolve this dilemma, and formulate a clearer theoretical identity for OSSTC— that not only respects the prior theoretical development of their ancestors (i.e., strategic alliances), but also further clarifies the nature of these particular collaborative efforts— I intend to lay the conceptual foundations of an inductive and eclectic theory of OSSTC, and position them vis-à-vis strategic alliances and informal cooperation.

---

<sup>3</sup> The seminal work of Hagedoorn (1993): "Understanding the rationale of strategic technology partnering..." provides a comprehensive overview of motives underlying formation of (strategic) interfirm technology cooperation.



C) To what extent does the business model concept explain open value creation processes?

A critical review of the business model (BM) literature (see Chapter 2, Section 5) shows, BM, as it seems, is still very corporate-centric, focused on firm-level sustainable competitive advantage, customer segment and revenue concentrated, and that it is focused on the internal structure of the firm.

Although the bulk of the literature tends to position the concept in the thinking of conventional strategic management– i.e., Porter’s five forces (1980a)– recently scholars have acknowledged that BM expands firm boundaries through adoption of an open strategy (e.g., Appleyard & Chesbrough, 2016) and that it is becoming more focused on value creation processes (Zott, Amit, & Massa, 2011). A BM, despite being perceived as a fuzzy concept (Zott et al., 2011; Teece, 2010; Morris et al., 2005) is a powerful tool to unlock the *latent value* from a piece of technology (Chesbrough & Rosenbloom, 2002). The recent literature (Zott et al., 2011; Teece, 2010) highlights the complexity of the ‘value creation process’ and ‘interconnectedness of exchange relationships and activities’ entailing a federation of players including the focal firm. Yet, BM does not seem to have become comprehensively attuned to open and distributed innovation concepts (e.g., OIP, user innovation, etc.) where value creation and capture processes are not readily situated within organizational boundaries. It certainly is poorly suited to explain the core and nuances of OSSTCs.

This is mainly the case because within OSS context, shared R&D and innovation process (RDIP) hosts a *unique* value creation process that includes numerous and heterogeneous actors (individuals, software/non-software public/private firms, communities, etc.) and encompasses different types of relationships which are embedded in open platforms. Such collective value creation process demonstrates a



complex set of interactions, and leadership challenges, to name a few. It defies traditional secretive software project management which is firm-centric (e.g., Apple Inc.) by making its development process openly visible and its artifacts publicly available over the web (Scacchi, 2002). OSS development process also challenges the conventional business and economics wisdom when it demands and motivates the private firms to share some of the important results of their private R&D in public domains (i.e., commons or open technology platforms).

Prior literature has placed the interrelationship between OSS development and user communities under the spotlight as these communities are an indispensable part of OSS development process (e.g., von Hippel, 2005). However, much less attention has been given to the role of the firms in the community, and how they can profit from their participation (van de Vrande et al., 2010)– with some exceptions (e.g., Dahlander & Magnusson, 2005). More importantly, literature has remained less focused on ‘how’, ‘why’, and ‘under what circumstances’ firms (not only software or OSS firms) should make a meaningful contribution to OSS development, enhance the success and sustainability of OSS projects and its RDIP in order to influence the technology trajectory, create positive network externalities, and shape the future standards of their technology domain. For instance, the investments that a firm or a consortium of organizations makes (i.e., boundary spanning collaborations) on developing a shared OSS platform (e.g., FFmpeg, Odoo, Liferay, etc.) in order to create new knowledge, can be considered as a unique value creation process which has pecuniary and non-pecuniary benefits for the participants.

Some authors have claimed that the vast majority of contributions within user communities are made by a few (Lakhani & von Hippel, 2003) where communities portray a core-periphery structure with a cohesive subgroup of core actors and a set of peripheral actors who are loosely connected to the core (e.g., Borgatti & Everett, 1999). However, we need to know more about the nature of value creation processes

to better explain why some remain, or position themselves as core to the OSS RDIP while others may shy away from an active engagement in the shared R&D process. Yet, some ‘highly professional’ participants<sup>4</sup> are not necessarily among the core development team, but they prefer to invest in the technology development process for they have incentives to do so. Further, what are the consequences for actors when they play different roles concerning their engagements with and resources endowments to OSS projects?

Thus, in line with bridging the conceptual gaps within the BM literature and progress the concept of open BM through investigating open, distributed and collaborative value creation process, I will study OSSTC and aim to unpack the value creation elements embedded in shared OSS RDIP.

### 1.2.2 Research problem from a practitioners’ perspective

Collaborative software development based on a wide range of OSS technologies has been increasing in the past decades (Linux Foundation, 2014). Today, OSS has become “the go-to platform” for organizations (e.g., Technologic Systems, Microsoft, etc.) to build software and technology products (Linux Foundation, 2016, p. 3). Based on a survey data gathered from more than 400 hiring managers at corporations, SMEs, government organizations, and staffing agencies worldwide as well as responses from more than 4,500 OS professionals, Linux Foundation report (2016) clearly illustrates the increasing “enthusiasm” and “professionalization” of OSS throughout IT industry. 59% of the respondents (i.e., hiring managers) claim that they are looking for OS talent in the job market. In fact, there is a new BM emerging where companies across different industries come together to share their development

---

<sup>4</sup> For instance, consider the case of some engineers from product engineering team of Savoir-faire Linux and their significant contribution to Linux kernel and FFmpeg projects (e.g., Savoir-faire Linux, 2017).

resources and build common OS technology platforms (i.e., tools and components) which they can later use to differentiate their own products and services— examples include AllSeen Alliances, Code Aurora Forum, OpenDaylight, OpenMDM, to name a few (see, for more details, Linux Foundation, 2014).

Subsequently, OSS solutions have been seriously discussed about, argued upon, adopted and even contributed to by private and public sectors. OSS is now at the center of attention among many European Union (EU) members where some already have their systems migrated to OSS versions while several others are seriously considering their action plan (see European Commission Joinup's website for complete up-to-date news )<sup>5</sup>. For instance, city of Munich— having in place an OS strategy focused on sustainability issue— has already completed their migration, LiMux project (i.e., “Die IT-Evolution”), to OS desktop (i.e., Ubuntu Linux operating system and OS applications) in which case 15,000 desktops run on OS (Hillenius, 2013).<sup>6</sup>

#### A) Macro level

At a macro level (i.e., government level), in Quebec, during the past years, adoption of OSS solutions by the Quebec government agencies has become a growing

---

<sup>5</sup> Joinup is a collaborative platform created by the European Commission and funded by the European Union via the Interoperability Solutions for European Public Administrations (ISA) Program. It offers several services that aim to help e-Government professionals share their experience with each other. They also support organizations to find, choose, re-use, develop and implement interoperability solutions. Retrieved from <https://joinup.ec.europa.eu/> on January 2016. For more information on the latest communications among private sector and governments, please refer to ‘Open Source Observatory’ which can be retrieved from <https://joinup.ec.europa.eu/community/osor/communications/all>

<sup>6</sup> LiMux – The IT evolution is a project by the city of Munich (third-largest city in Germany) to migrate their software systems from closed-source, proprietary Microsoft products to free and open-source software. The project was successfully completed in late 2013, which involved migrating 15,000 personal computers and laptops of public employees to free and open-source software. The description of LiMux project is retrieved from Wikipedia website on January 22, 2016 available at <https://en.wikipedia.org/wiki/LiMux>.

controversial issue.<sup>7</sup> On the one hand, switching to OSS solutions is advantageous as the government can save huge amounts of taxpayers' money and ensure Quebecers to enjoy higher security and freedom. For example, only "En 2012, le gouvernement a choisi de renouveler une série de licences de gré à gré, pour un coût total évalué à 1,4 milliard \$. S'il avait plutôt opté pour des logiciels libres, il aurait pu réaliser des économies d'au moins 19 %, soit 265 millions \$" (Therrien, 2013)<sup>8</sup>. Similarly, proponents of protection of citizens' information and digital privacy view adoption of OSS solutions at the heart of "la stratégie culturelle numérique" (i.e., the strategy of digital culture).<sup>9</sup> On the other hand, the government is left with little empirical research evidence in the context of Quebec, and Canada for that matter, in order to be confident about the capacity and reliability of OSS projects to depend upon. The government has also made large path-dependent investments with large corporations (e.g., Microsoft, IBM) which has led to proprietary lock-in situation (Deglise, 2013).

Therefore, it is high time to look into the black box of OSSTC embedded in shared OSS R&D and innovation projects to provide an unbiased explanation about their internal workings and mechanisms. The findings of this empirical investigation have implications for policy makers, as they help them gain a finer-grained understanding of the subject matter. Particularly, this study is useful in the context of Quebec where

<sup>7</sup> This has also led to some lawsuits against government actions and decisions with regards to adoption of OSS solutions through contracting with private sector's OS firms. For an example, please read the following reports. Nowak, P. (2008, Aug 27). Quebec government sued for buying Microsoft software. *CBCnews*. Retrieved from <http://www.cbc.ca/news/technology/quebec-government-sued-for-buying-microsoft-software-1.762879>. Nowak, O. (2010, June 3). Quebec broke law in buying Microsoft software. *CBCnews*. Retrieved from <http://www.cbc.ca/news/technology/quebec-broke-law-in-buying-microsoft-software-1.962964>.

<sup>8</sup> In English: "In 2012, the government has decided to renew a series of licenses, for a total cost estimated at \$ 1.4 trillion. Had the government opted for OSS, they could accomplish savings of at least 19%, or \$ 265 million"

<sup>9</sup> For a reference, please read: a) Le Devoir. (2014, June 18). *Mutations numériques «On ne subit pas l'avenir, on le fait»*. Retrieved from <http://www.ledevoir.com/societe/science-et-technologie/411245/mutations-numeriques>; b) le manifeste (2014). pour un Québec numérique libre et ouvert. Information website: <http://www.notreavenirnumerique.net/>; and c) Communautique (2010, October 16). Manifesto: A Digital Framework for Quebec. Retrieved from <http://www.communautique.qc.ca/reflexion-et-enjeux/internet-citoyen/manifesto-digital-framework.html>.

the government is in a *transition phase* and challenging with several issues such as OS licensing (Hillenius, 2016). Policy makers need to be better informed of the *nature of OSS value creation process*. This will consequently help them to ensure that their efforts about regulating the OSS licensing based on the government needs will not restrict the openness of OSS development methodology.

#### B) Meso-Micro level

At meso-micro level (i.e., group or firm level involving individual developers), effective OSS development requires meaningful collaboration among firms, groups of people, individuals, and developer communities. Many OSS projects carry a lot of potential, but the problem is that the communities creating them are not necessarily cohesive. With different cultural values, economic incentives and even technical backgrounds, the slightest disagreement among developers (from firms, large corporations, joint networks, or even simply individuals) is often enough to provoke an abandoning of the project and independent development by a subset of developers (i.e., *project forking phenomenon*<sup>10</sup>). Forking is a double-edged sword. It can kill projects, and it may sometimes lead to a new flourishing project. However, despite its ubiquity within OSS ecosystem, forking is mostly viewed as something negative and it is frowned upon. Our preliminary pilot research shows that if a project is lacking adaptation capability in order to respond to real market needs, it will finally become an orphan and will eventually disappear due to a dwindling user-base. So far, OSS has been mainly perceived as social phenomenon comprising a network of individuals who have gained mutual trust and respect through meritocracy and reputation effect. But in view of strategic and commercial OSS which competes with proprietary closed software products manufactured and packaged by large

---

<sup>10</sup> According to Kelty (2008, p. 136): “*Forking* generally refers to the creation of new, modified source code from an original base of source code, resulting in two distinct programs with the same parent”.

corporations, the role of firms in developing OSS platforms and technologies is more under spotlight specifically when they can influence the success and sustainability of the projects through their collaborations and resources endowments.

In fact, this is due to the fact that OSS is dynamic software always unfinished and constantly in the state of flux. This does not mean that OSS cannot be used effectively because it is changing and being improved upon continuously. It simply means that the users (and in sense of strategic and commercial OSS mainly firms from different industries) need to be aware of its *evolutionary nature* and remain connected to the core software hosted by community around the project in order to ensure interoperability. Therefore, *take-and-give-back* captures the nature of interactions among actors (under certain conditions). Users adopt a solution, work with it, and share back their experiences with the software with the community of users through a number ways. The sharing-back ranges from giving a simple feedback by mentioning how the software reacts on their hardware (does it crash?), highlighting a glitch in their systems, to more complicated contributions such as writing the patch, submitting it to the community for a review, and getting involved in its revising process until the clean code is integrated into the core. However, to engage in such interactive process, one shall be familiar with several issues, and capable of performing several tasks. Put simply, firm's IT department representatives need to be aware of core issues such as *how* and *why interdependence* among collaborators influence technical and commercial success of OSSTC and OSS shared RDIP. Furthermore, they need to know *what capabilities* are required to play an active role and if creating such capabilities can become economically justified before their top management.

Although OSS service providers (i.e., OSS firms) have mastered these issues, still a large group of non-software firms which form a big portion of OSS user base seem to be only aware of the tip of the OSS RDIP's ice berg. The clients may have heard that

adoption of OSS solutions saves their IT department some costs, but they may have been ignorant or unaware of the fact that if a firm relies on an orphan project that lacks continuous supports and further development, the issue of cost is no longer an advantage. It is because an orphan project is not an effective problem solver. Thus, in this research, we shed light on vital details associated with OSS RDIP, and explain the core issues (e.g., nature of leadership in OSSTC) in order to assist firms to make better informed decisions with regards to choosing *what role*, and *to what extent*, they want to play when engaging in OSS adoption process (i.e., migrating toward OSS solutions).

### 1.3 The design of the research questions

I believe arriving at valuable research questions is a research project on its own. In this research, two processes have influenced the research questions' development: a) the literature review process; and b) the pilot study. The former informs me about the 'ins and outs' of the key notions such as OI, BM, strategic alliances, and OSS projects (from an academic perspective). It also helps me form some opinions about the interconnections among these topics. The latter, however, influences my understanding of OSS technology and its development process more significantly from practitioners' perspectives. My five preliminary interviews with industry practitioners such as R&D managers of software and OSS firms located in Montreal, Waterloo and Ottawa, as well as a high rank policy maker from Quebec City directed me to formulate questions that are more specific. These questions not only correspond to the gaps I have identified based on literature review but also they address queries that practitioners may have.

For instance, in the OSS literature, Weber (2004, p. 216) highlights "the role of the 'customer' in the production process, specifically as it plays out within business

models” a central conceptual question to be further studied in the context of OSS RDIP. Later, in my pilot interviews with the founder as well as the chief technology officer of Savoir-faire Linux Inc., I became engaged in discussions that would further lead to importance of the interconnection between OSS communities, OSS firms and the enterprise clients. However, the interviewees mention that the client’s role and its significance is a matter of investigation and that it is subject to change as each case could be a different one.

Thus, such preliminary investigations (both conceptual and practical) and the information load they carry tend to influence my judgment and direct me towards formulating questions that are more context-specific. Some may view this as a source of investigator’s bias. I personally believe that this cannot be necessarily a downside, although it can limit a researcher’s window of openness to a certain extent. The fact of knowing that there may be another element such as enterprise clients involved in collaborations embedded in OSS RDIP can affect the dimensions of my thoughts. However, since I do not possess any specific knowledge of how, why, and under what circumstances such interconnectedness influences OSS RDIP and its dimensions, I remain confident that I am entering the field study with a blank slate. Below, I propose a set of primary and secondary questions (see Table 1.1). These questions play the role of the catalyst for my exploratory investigation.



Table 1.1. Research Questions

Item no.	Primary and secondary research questions
<i>Collaboration-driven questions</i>	
PQ1	What is the nature of “OSS value creation processes”; and what key factors shed light on its complexities?
SQs	a) Who are the main independent actors; and how are they interrelated to one another, i.e., what is the nature of their interdependencies?
	b) What are the underpinning mechanisms, relationships, capabilities, and challenges that form open, distributed and collaborative value creation process?
	c) How or in what significant ways, do the actors’ technological collaborations lead to enhancing the success and sustainability of collaborations and shared R&D and innovation processes?
	d) How do openness and open collaboration influence project success and sustainability?
	e) How are the second or third-partner dependencies managed and led in order to be effective?
<i>Market (client)-driven questions</i>	
PQ2	How may organizational users (e.g., enterprise clients) influence collaborations and OSS RDIP?
PQ3	How may organizational users (e.g., enterprise clients) affect sustainability of OSS RDIP?
<i>Leadership-driven questions</i>	
PQ4	a) What are the key capabilities of a leader? How do they influence the governance of OSS technological collaborations and projects?
	b) How is leadership viewed when there is a plethora of independent actors? Does leadership of OSS technological collaborations take on a different color by leaning on different sorts of capabilities and demonstrating a unique set of performances?
PQ5	a) What organizational challenges are imposed by involving the third actor – e.g., clients – into the innovation process? How can they be addressed properly by OSS firms and community of developers to make sure they will not negatively influence success and sustainability of OSS projects. In other words, how can OSS firms and communities realize the clients’ potentials in spite of the challenges associated with their utilization?

Note: PQ: Primary question; SQs: Secondary questions

#### 1.4 Research methodology

Three main considerations have influenced my decision to choose qualitative research methodology, drawing upon a mix of three variants of it: case study (Yin, 2009); grounded theory approach, GTA (Glaser & Strauss, 1967; Strauss & Corbin, 1990); and, participant-observation (Becker & Geer, 1957; DeWalt & DeWalt, 2002). These considerations include the nature of the research problem, study purpose and type of research questions. Thus, in order to realize this research project, I must adopt the qualitative approach.

I intend to explore, describe, and explain the core issues associated with OSSTC and OSS shared RDIP. This includes studying the experiences of individuals (representing their organizations and themselves) actively involved in OSS RDIP—their sense making, collaborative knowledge sharing and knowledge creation activities, their strategies, as well as their decision-making. My goal is not to prove or disprove any hypotheses, rather to inductively develop theoretical constructs, a gestalt theoretical framework, and core propositions that explain how actors and constructs are interconnected as well as how different variables influence the proposed interrelationships. Furthermore, based on follow-up discussions, I will propose several testable hypotheses to guide further empirical, theory testing, and quantitative research approaches.

Yin (2009, p. 18) characterizes case study as “an empirical inquiry that investigates a contemporary phenomenon in depth and within its real-life context, especially when the boundaries between phenomenon and context are not clearly evident”. The *case study method*, therefore, is suitable for examining a phenomenon such as OSS RDIP that is very context-dependent where boundaries between firms, communities and individuals become very blurry and porous.

Adopting *participant observation* in combination with case study is also advantageous because it is a powerful analytical tool and a data collection method that enhances the quality of data collection, and data interpretation (DeWalt & DeWalt, 2002). In addition, the *grounded theory approach* (as described and commented on in Chapter 4) is an appropriate approach because it aids the researcher to systematically follow an emic or inductive approach to data collection, analysis and theory building (Strauss & Corbin, 1990).

To collect data, I conducted the participant-observer and the in-depth main case study in *Savoir-faire Linux Inc.*'s Montreal headquarters for one year. All along this process, relying on purposive and snowballing (theoretical) sampling techniques, I have extended the number and nature of my cases to include different types of enterprise clients, OSS firms, and OSS communities. In total, I have conducted 40 in-depth and focused interviews with chief technology officers, R&D managers of OSS firms and projects, OSS community project leaders and highly active OSS developers who have held maintainer's positions in different projects. I have also talked informally, face-to-face, with a number of developers on OSS-focused occasions such as workshops and seminars being held at *Google Montreal*, *Computer Research Institute of Montreal* (CRIM), *HackerNest Montreal* at Notman House, to name a few. The interview contents are then analyzed based on the systematic procedures such as constant comparative method (CCM) explicated in several major works including Glaser and Strauss (1967), and Strauss and Corbin (1990).

Finally, since 'generalizability' is a major concern to most researchers to the extent that it may cast doubts on the usefulness of the research results (e.g., see Lee & Baskerville, 2003), it is necessary to discuss my research findings in light of the generalizability matter.

First, Lee and Baskerville (2003, p. 221) highlight that “many IS researchers, both quantitative and qualitative, have restricted themselves to just one particular notion of generalizability- namely, a statistical, sampling-based notion”. Such narrow view of the concept has caused IS researchers to even impose “this particular notion even outside the bounds of statistical, sampling-based research” (Lee & Baskerville, 2003, p. 221). The present research’s findings, however, fall under the “Level-2 inference” building or “analytical generalization” notion which, according to Yin (1994, p. 31), is generalizing from case study findings to theory. Thus, following Yin (1994), and as emphasized by Lee and Baskerville (2003, p. 222), “statistical generalizability is inappropriate as a measure of the quality of case studies because they involve not only a different form of inference (inference from case study findings to theory rather than from a sample to population characteristics [i.e., Level-1]), but also inference at a different level (Level 2 rather than Level 1)”.

Second, Lee and Baskerville (2003, p. 232) claim, “...the outputs of generalizing (the “generalizing notions”) can be either theoretical statements or empirical statements, and the inputs to generalizing (the “particular instances”) can be either theoretical statements or empirical statements. Thus, based on their four classifications, this research falls under the second: ET; i.e., “generalizing from description to theory. This involves the researcher’s job to generalize from empirical statements to theoretical statements (Lee & Baskerville, 2003, p. 235). Such form of generalization, according to these authors, corresponds with Yin (1984, 1994)’s synonyms for generalizing from *empirical to theoretical statements* such as: “analytical generalizability”, “Level-2 inference”, and “generalizing to theory”. In fact, generalizing from case study findings (e.g., empirical descriptions and rich details) to theory is a form of generalizing from empirical to theoretical statements. The outcomes of this research that are constructs, the relationships among them in form of propositions and hypotheses, thick descriptions, as well as models are indeed the theoretical statements.

### 1.5 The significance and relevance of the research

This study is significant because it addresses the questions that are both relevant and important to academics, policy makers and industry practitioners who are interested in OSS technology. The theoretical field of OSS is a rich research domain but it has not reached its critical mass yet. More particularly, as OSS is a socio-economic technology it has been studied from different perspectives and in an interdisciplinary fashion, therefore, its research field is quite fragmented. Thus, my work is important as it provides a multifaceted understanding of the subject matter. It studies OSS and its relation to OI, strategic alliances, and BM.

Primarily, by untangling OSS from OI conceptual camp, and delineating the two literatures, I have progressed our understanding of both OI and OSS. Academics do not necessarily need to rely upon insights from OI framework to study and theorize OSSTC. This is because OI is not a very specific and comprehensive explanatory tool to encompass the multiple dimensions of such intricate and multifaceted technology. For instance, considering that OI is still a work in progress (e.g., Trott & Hartmann, 2009), and that the “Theories that can be aligned with open innovation [e.g., RBV] still have to be modified to grasp open innovation” (Vanhaverbeke & Cloudt, 2014, p. 274). We cannot put all our eggs in one basket, and rely much on OI theory to advance our detailed understanding of OSS boundary spanning collaborations.

Next, by looking deep into OSSTC and OSS RDIP through adoption of the bottom-up approach, I will provide a finer-grained understanding of key minute conceptual building blocks as well as core categories which both academics and practitioners may find very challenging to explain as well as highlight their interconnections through a gestalt framework. For instance, breaking down the two concepts of success and sustainability of OSS projects or OSS shared RDIP can benefit academics with their deductive research designs to test OSS-related hypotheses with

higher degree of precision. In order to design a survey study, researchers can create items (variables) that are discussed in this research, and therefore increase the content validity of their survey tools. Practitioners (e.g., OSS firms' sales, communications and marketing departments, and OSS community administrators) can also integrate this research's findings and discussions into their information sessions, and creation of marketing and business intelligence tools in order to communicate their message to commercial clients more effectively and justify why they are seeking their active collaboration and long-term commitment with OSS projects. Even, IT departments of clients can benefit from these research findings to further justify their request for more support (resources, empowerment, etc.) from their top management on occasions their firm's IT strategy involves migration toward OSS technologies.

In addition, this research positions OSSTC as a new breed of conventional strategic alliances and distinguishes it from informal cooperation. Therefore, academics who are interested in theorizing about either strategic alliances or OSSTC can benefit from the theoretical perspectives developed in Chapters 5 and 6 in order to be more specific in their explanations of the collaboration phenomenon and typology development efforts. Similarly, OI and OBM researchers can have a more detailed look into the black box of open, distributed and collaborative value creation processes within OSS context and enrich their OBM theory development efforts by drawing on this research's results.

Furthermore, policy makers who always need more recent and detailed research on OSS technology may find this research's findings interesting and relevant to their decision making. They must to know why, how, and to what extent they need to be engaged with OSS communities and projects. Even some government bodies that serve their own downstream multiple clients can benefit from the results of this research. In fact, the framework and core propositions developed in Chapter 5 can extend policy makers' *gestalt* as well as *detailed understanding* of OSSTC and OSS

shared RDIP. Once they know about micro and macro factors involved in the OSS development process (particularly through a simple and plain language and examples used in this dissertation), they can enter decision making sessions with a broader perspective and knowledge of the repercussions of their decisions on the viability of the upstream OSS projects that they are planning to integrate into their information systems.

Last but not least, it is also important to pay attention to the consequences of *not* doing this research. Prior important works on OSS (e.g., Weber, 2004; Kelty, 2008) – mainly based on interviews with developers of two cases (Linux and Apache) have explained key issues such as individuals’ motivation to engage in software development, the complexity of OSS, and the coordination among large numbers of developers among other things. Yet, they have paid little attention to *OSS as a commercial and profit generating software which is developed on an open and collaborative technology platform* (OCTP). Such open platform can include a diverse demographics and for this reason it is important to know how different groups of participants such as ‘enterprise clients’ as users can be factored in the equation of OSS firms plus community of developers in developing, steering, sustaining and profiting from OSS projects.

As a case in point, Weber (2004, p. 216) highlights “the role of the ‘customer’ in the production process, specifically as it plays out within business models” a central conceptual question to be further studied. Weber (2004, p. 216), cites Eric Allman (2001)’s argument that: “from a business perspective unsophisticated customers are free riding in the classical sense and tend to degrade the value of software”; and he further claims that this may be the case because the client’ “business model has not figured out how effectively to channel the knowledge flowing in the other direction into the software rather than letting it be dispersed”. To not know *how* and *why* OSS firms and community leadership deal with ‘commercialization challenge’ of OSS

technology impedes our understanding of sustainability of the OSS collaborations. In fact, the puzzle of sustainability (as evident in cases of BigBlueButton and Tiki Wiki CMS Groupware) can be better resolved by knowing more about the interplay between long/short-term profit making, and sustainable technology development.

For these important reasons and many more, this research intends to provide value for academic and industry readers.

## 1.6 Limitations

This research contains a number of inherent limitations that have been beyond the bounds of my direct control and have had an impact on it. The major limitations in brief are mentioned below, while they are explained in more detailed in the final chapter (i.e., Chapter 7, Section 7.5).

The first major shortcomings are classified under '*research logistics and socio-cultural related issues*'. These include the requirements for strict confidentiality from private enterprises, anonymity, and privacy of the participating individuals. Furthermore, issues like availability (and not necessarily willingness) of managers (e.g., R&D project managers, Topnotch coders, etc.) in combination with time differences for those stationed in Europe or online accessibility for those stationed in other provinces in Canada provided challenges in the data collection process. Lastly, being an outsider to OS world that –having its own unique socio-cultural issues– has been an important factor.

Second, the issue of the '*researcher's theoretical sensitivity*' played both an advantageous and disadvantageous role. My low theoretical sensitivity level has helped me to remain an impartial investigator in the course of this research. It has



been also a limitation, specifically in high-tech arena such as software world, and in particular, OSS world that is full with jargons and technical terms. Finally, there have been challenges of higher level of abstraction in the process of categorization and conceptual groupings activities associated with GTA and inductive theory building.

Third, I cannot claim that the theoretical statements that I have developed in this work will remain valid beyond the observed cases. Lee and Baskerville (2003, p. 236) mention: “a theory generalized from the empirical descriptions in a particular case study has no generalizability beyond the given case”. However, as difficult a task to accomplish generalizability is, Lee and Baskerville (2003, p. 236) acknowledge that “this particular lack of generalizability is not only a feature of qualitative studies, but also statistical, sampling-based studies”. By comparing the notion of generalizability between case research and statistical research, Lee and Baskerville (2003, p. 236) draw a parallel between generalizing beyond a given field setting in the former and generalizing beyond the given population in the latter where “sample points may be generalized to sample estimates of population characteristics, but certainly have no generalizability beyond the given population”.

Having acknowledged the weaknesses of this research related to generalizability issue, other limitations have not critically inhibited this empirical investigation mainly due to adopting the *participant observation method* carried out during a yearlong industry internship program that I have completed in OSS industry.

### 1.7 Dissertation outline

The dissertation is comprised of seven chapters. *Chapter 1* introduces the topic, highlights the research problems from academics’ as well as practitioners’ perspectives, poses the primary and secondary research questions and the

methodology through which this research is conducted. It further emphasizes why this research is important and relevant to academics, and practitioners. And finally, it sheds light on major limitations of this study and concludes with this section– i.e., the dissertation's outline.

*Chapter 2* provides a critical review literature of OI and its related paradigms. It describes what OI concept is, and discusses openness as it lies at its core. It further provides a review of theoretical developments on OI including open strategy, OI business models, OSS business models and other major theoretical developments related to OI framework.

*Chapter 3* defines, describes, and discusses origins and significance of OSS as an appropriate case to study open and collaborative technology development. It also identifies the inherent particularities of OSS technology. It further reviews the theoretical perspectives on OSS development process and highlights the most valid and important empirical findings related to OSS research domain. It emphasizes the transformation of OSS into a more strategic and commercial option that competes with closed proprietary software solutions. Finally, it highlights the problematic areas and stresses why we need to adopt an inductive approach to study OSS technological collaborations.

*Chapter 4* describes the research methodology and justifies the underlying reasons, the rationale, for choosing a qualitative research approach. It further highlights and explains why a combination of three variations of qualitative methodology is employed in this research. This chapter also explicates the research design by defining the unit of analysis, the major case and minor cases, sampling and data collection, as well as data analysis methods. It further discusses the issues of validity, reliability, generalizability, and ethical considerations, and finally closes the discussion by highlighting the methodological contribution.

*Chapter 5* and *Chapter 6* are the fruits of this research. *Chapter 5* provides the major and minor findings about an inductive theory of sustainable OSS shared R&D and innovation process. It introduces the theoretical model of OSSTC, and identifies the key factors influencing the success and sustainability of collaborative OSS RDIP. It also proposes and explains a set of core propositions, discusses them and recommends testable hypotheses to guide further empirical investigations. Therefore, the theoretical model, propositions, and hypotheses, together, put the main conceptual categories, subcategories and their conceptual building blocks in perspectives—therefore, offering a gestalt overview of the inductive theory developed in *Chapter 5*.

*Chapter 6* critically reviews the literature of strategic alliances, and the major management and economics theories employed in that context with the goal of better understanding and identifying the position of OSSTC in relation to conventional strategic alliances. Furthermore, building on the main findings of *Chapter 5* and using the insights from the literature reviews (*Chapter 2* and *3*), *Chapter 6* develops open and collaborative technology platform as an alternative perspective of OSSTC.

*Chapter 7*, the final chapter, concludes this research endeavour by highlighting the theoretical contributions as well as managerial implications of the results of this study. It further identifies the major study limitations and how they have been overcome. Lastly, it recommends several avenues to extend this research.

## CHAPTER II

### A CRITICAL REVIEW OF THE THEORETICAL PERSPECTIVES ON OPEN INNOVATION

#### 2.1 Introduction

This chapter analytically focuses on the theoretical perspectives on OI concept. I first discuss the changing landscape of innovation and, drawing on the relevant literature, argue that although ‘openness’ is an essential condition for innovation and particularly adopting an OIP, it is not sufficient to succeed and compete in the industry. Next, I review and discuss major theoretical and empirical works conducted in the field of OI and highlight several of their shortcomings. Further, I explain how business model concept fits within OIP and where it falls short of catching up with the concept of OI. In addition, I touch upon open strategy, and open business model concepts as two enablers to adopt OIP. These discussions further lead to remaining caveats in the open business model research and its limitation to properly explain open and distributed value creation processes. Later, I try to be more objective and take a critical perspective toward OI. I, therefore, discuss its major weaknesses and reiterate the existing claim that OI is not a fully-fledged theory yet.

#### 2.2 Open innovation concept and the changing landscape of innovation

*Innovation* is perceived as a necessity to a firm’s survival and growth. Innovation scholars, through challenging the oversimplified “linear” view of innovation (process), have highlighted several key *realities* about its inherent characteristics. They view innovation as “flexible” (Niosi, 1995), “interactive” and “complex, uncertain, somewhat disorderly, and subject to changes of many sorts” (Kline &

Rosenberg, 1986, p. 275; von Hippel, 1988). They have further emphasized innovation as a “collaborative enterprise”– one that rests upon the blurring boundary of the firm (e.g., Gomes-Casseres et al., 2006; Lindsey, 2008).

In 2003, Chesbrough (2003a: XXIV) coined the term OI to differentiate the changing landscape of innovation in modern era. He defines OI as “a paradigm that assumes that firms can and should use external ideas as well as internal ideas, and internal and external paths to market, as firms look to advance their technology”. Later, Chesbrough and Bogers (2014) refine the definition of OI to further clarify it and bring conceptual uniformity to its field of research. They define OI as “a distributed innovation process based on purposively managed knowledge flows across organizational boundaries, using pecuniary and non-pecuniary mechanisms in line with the organization’s business model” (Ibid., p. 17). Having positioned the ‘new’ concept in the literature as the stark opposite of the ‘closed’ innovation paradigm (CIP), Chesbrough (2003c, p. 38) highlights *six fundamental principles of OI* (Table 2.1).

To state the obvious, OI has enjoyed an unprecedented upward trend in terms of popularity measured through academics and practitioners’ publications (Chesbrough & Bogers, 2014). However, this does not mean that OI concept is *a bed of roses!* Quite the contrary, OI suffers from three major criticisms that I will discuss later in this chapter.

Historically, large firms have been mostly dependent on internal R&D to make successful innovations happen. In fact, based on the tenets of the resource-based view (RBV), firms gain and sustain competitive advantages by deploying valuable resources and capabilities that are inelastic in supply (Wernerfelt, 1984; Barney, 1986, 1991; Peteraf, 1993). Therefore, large internal R&D laboratories have been viewed as a strategic asset and motor of innovation acting as an entry barrier for

potential rivals (van de Vrande et al., 2009). Many firms used to profitably engage in internal R&D and their preference has been to use the proprietary R&D model “where internal R&D activities lead to products that are developed and distributed by the firm” (Chandler, 1990). While the proprietary and closed organizational design cannot be viewed as an absolutely obsolete model today, it can, in many cases, have its own limitations and renders itself less favorable concerning innovation process.

Table 2.1. Contrasting Principles of Closed and Open Innovation

Closed innovation principles	Open innovation principles
The smart people in our field work for us.	Not all of the smart people work for us* so we must find and tap into the knowledge and expertise of bright individuals outside our company.
To profit from R&D, we must discover, develop, and ship it ourselves.	External R&D can create significant value; internal R&D is needed to claim some portion of that value.
If we discover it ourselves, we will get it to the market first.	We don't have to originate the research in order to profit from it.
If we are first to commercialize an innovation, we will win.	Building a better business model is better than getting to the market first.
If we create the most and best ideas in the industry, we will win.	If we make the best use of internal <i>and external</i> ideas, we will win.
We should control our intellectual property (IP) so that our competitors don't profit from our ideas	We should profit from others' use of our IP, and we should buy others' IP whenever it advances our business model.

\* This maxim first came to my attention in a talk by Bill Joy of Sun Microsystems over a decade ago. See, for example, A. Lash, “The Joy of Sun,” The Standard, June 21, 1999, <http://thestandard.net>.

In line with this argument, Chesbrough's (2003a, b, c) OI model posits that the landscape of innovation; more precisely closed innovation, has been changing through the emergence of a number of “erosion factors” in the twentieth century. Chesbrough (2003c, p. 36) enumerates the most important of these factors: 1) the “dramatic rise in the number and mobility of knowledge workers” which imposes

firms the challenge of controlling their proprietary ideas and expertise; 2) the “growing availability of private venture capital” as it stimulates the commercialization of ideas that have “spilled outside the silos of corporate research labs” through providing the finances.

Others (e.g., Mina et al., 2014, p. 853) have also emphasized such shift in the grand landscape of innovation regime where firms are developing “more outward-looking strategic approaches to research and development to source at least some knowledge of potential value from the broader environment in which they operate”. Influencing factors include vertical disintegration pressures (Langlois, 2003), modularisation and outsourcing (Prencipe et al., 2003; Sturgeon, 2002), the growth of specialised technology markets (Arora et al., 2001; Brusoni et al., 2001) and difficulties in appropriating internal investments in intangibles (Chesbrough, 2003b). These factors encourage firms to develop more open and porous organizational boundaries by adopting the OIP. Such paradigm allows for more inclusion of external knowledge and paths to markets in the innovation management process.

The open approach to innovation also benefits firms in a number of ways. It enables firms to better adapt to the dynamic market needs, pool their resources together and share the costs and risks associated with R&D among partners, and enjoy higher rates of commercial returns, to name a few (Chesbrough, 2003a; 2006b). Partly due to its imperative and partly because of its merits, increasing number of firms has embraced OIP and its embedded innovation strategies in recent years (Hagedoorn, 2002; Kirschbaum, 2005; Huston & Sakkab, 2006; Roijakkers & Hagedoorn, 2006).

Irrespective of its advantages, OIP is also fraught with various challenges associated with disadvantages of openness. The selective openness associated with the OI strategy and OI Business Models (OIBM) induces a certain degree of revealing of the firm’s internal knowledge to the external environment and sharing strategic resources

with the goal of collecting the returns through different mechanisms. Thus, since openness lies at heart of OIP, I discuss it in more detailed.

### 2.3 'Openness': The essential condition to adopt open innovation paradigm

Chesbrough and Appleyard (2007, p. 57) define openness as "the pooling of knowledge for innovative purposes where the contributors have access to the inputs of others and cannot exert exclusive rights over the resultant innovation". Some aspects of the innovation process are open and others may be closed (Chesbrough et al., 2006b), however, researchers tend to view openness as a matter of degree rather than a black-and-white or on-and-off situation (see e.g. Dahlander & Gann, 2010; Lazzarotti, Manzini & Pellegrini, 2011; Drechsler & Natter, 2012). For example, Dahlander and Gann (2010, p. 703) maintain that "if we accept that openness is a continuum, a non-controversial argument in the open innovation community, then we can seek to advance a greater understanding of benefits and costs of openness".

As openness is the necessary condition to adopt OIP, research on openness has been continuously under spotlight in the OI literature. For example, Lichtenthaler (2008) has paid close attention to the implications involved with the firm's emphasis on radical innovation. He finds that as a firm's degree of emphasis on radical innovation increases, the firm's degree of openness seems to rise accordingly. This is even more obvious when the degree of external technology commercialization is under spotlight.

Further, Lazzarotti et al. (2011), studying Italian manufacturing firms, find out four different OI models (*open innovators*; *closed innovators*; *integrated collaborators*; and *specialized collaborators*) with respect to two variables which represent the 'degree of openness' within a firm's BM. Lazzarotti et al. (2011) further classify the extent of openness into two: 1) *partner variety* (the number and type of partners); and



2) *innovation phase variety* (the number/type of phases of the innovation process which are actually open to external collaborations). Next, the authors investigate how four firm-specific factors (R&D intensity, size, approach to innovation, and organizational and managerial actions) affect the degrees of openness thereby leading to creation of different OI models.

Having perceived openness as a matter of degree, several scholars have focused their attention on the '*impacts of openness*'. For example, Fey and Birkinshaw (2005) have been successful in showing that the greater the openness to new ideas in a firm, the higher the firm's R&D performance. While others like Laursen and Salter (2006a), based on a large-scale empirical study, maintain that there is an optimal degree of openness with regards to a firm's external search strategies. Following this line of argument and the premise that openness is a matter of degree (i.e., openness can be viewed from a continuum perspective), Drechsler and Natter (2012) use contingency theory (Hambrick, 1983) to empirically find out about the factors underlying the degree of openness. Contingency theory (Hambrick, 1983) emphasizes the importance of knowledge of antecedents to gain a better understanding of a particular business orientation, i.e., the decision on how open a firm should be.

The work of Drechsler and Natter (2012) is of particular importance partly because it is a pioneering research in its kind, and partly due to its research framework. Based on this work, the main factors that relate to the degree of openness (i.e., closed vs. various levels of openness) are four: 1) a *firm's innovation strategy*; 2) *scarce firm resources*; 3) *the appropriability regime*; and 4) *market dynamics* (Ibid., p. 439). These four factors are structured at two levels based on *firm-specific or internal* factors as well as external or environmental factors. The results of this study show that three main factors stop firms from opening up, and therefore remaining in the closed innovation zone. These factors are: 1) lack of market and technological knowledge (knowledge gaps); 2) ineffective intellectual property (IP) protection

mechanisms; and 3) competitor's threats such as market entries and imitation. Further, the most important factors that impact a firm's strategic decision to opt for openness are two: 1) a firm's need for financial funding in innovation; and 2) the effectiveness of a firm's IP protection mechanisms. When higher degrees of openness are concerned, Drechsler and Natter (2012) find that, still, scarce financial resources and effective IP mechanisms are drivers for opening further up.

Finally, long before OI scholars emphasize the notion of openness in terms of its degree, antecedents, or impacts, Niosi (1995) – conceptualizing innovation process (i.e., R&D) as a flexible phenomenon which heavily relies on interfirm collaborations – find that small and medium-sized enterprises (SMEs) tend to more frequently rely upon interfirm R&D collaborations when compared to the large firms. Across different industries (e.g., electronics, biotechnology and advanced materials), as Niosi (1995) explains, “size” appears to be one of the key explanatory variables underpinning the collaborative behavior of firms. For instance, SMEs in electronics sector, due to resources (pecuniary and human capital) shortages view collaborative R&D as a conduit to obtain “economies of scale” so much so that either they form larger number of collaborations and/or they allocate a larger portion of their R&D effort (i.e., collaborations as percentage of R&D expenditures). In a nutshell, the smaller the firm is, the need to become more open and reach out for resources seems to be more pressing.

These results inform us of different conditions under which a firm decides to open up or remain closed as well as the conditions under which a firm decides to open up to a higher or a lower degree. The first preventing factor shows that a firm needs to have certain capability(ies) to primarily understand what it needs in terms of new technological knowledge appreciation and comprehension. This shows how much the firm-level innovation capability and absorptive capacity theories (e.g., Cohen & Levinthal, 1990) are still relevant when issue of openness is at stake. Second, IP

regime is still a strong mechanism for value capture and the majority of firms are unable to develop new strategies (other than IP) in order to appropriate the value of their innovations in an OI regime. Thus, this lack of appropriation capability holds firms back from further adopting an OIP and engage in the opening-up process. Third, financial needs provide a strong motivation to open up the firm's innovation process. This suggests that RBV of the firm is still a relevant theory which can be relied on in the field of OI particularly when the strategic decision of open vs. close is at stake.

To continue, in a more recent study, Felin and Zenger (2014) acknowledge that, based on 'firm-level aggregates', the past literature on OI has made a rather strong and positive case for more open governance forms (for example, alliances and joint ventures, firm boundary permeability, etc.) in favor of a firm's innovation outcomes. However, the authors shift the spot light on the *caveat which encompasses governance solutions at the micro-level*; meaning that if a firm follows prescriptions based on aggregates, it may run the risk of making governance (open versus closed governance structure- i.e., management of innovation process) mistakes at the micro level. In so highlighting the issue, authors emphasize on the 'problem' which derives innovation as the focused process, and tend to argue that the decision to opt for open vs. closed governance structure must be made so as to strike the right balance between *problem type and governance form*—i.e., problem-governance match making. The premise to this analysis is that "differing problems, in essence, demand differing approaches to solution search", according to Felin and Zenger (2014, p. 916). In this tentative theory, the manager is the principal actor who "seeks to effectively govern, manage and organize problem solving associated with innovation" (Ibid., p. 915).

Felin and Zenger (2014) develop a matrix in which there are two dimensions: 1) problem complexity; and 2) hidden knowledge. Further, authors identify two search dimensions: 1) the source of the direction guiding the search (simple, decentralized

trial and error, or theory-driven search); and 2) the mechanism which solicits participation in search (centrally identified and selected vs. self-nominated and selected). Therefore, problems with different level of complexity are different in their need for knowledge exchange and knowledge discovery and so they need to be matched with their own governance forms.

As a case in point, in a situation where problems are complex and knowledge required to solve the problem is hidden and dispersed, a firm may use users' communities. Android Inc., once a private firm which was later acquired by Google Corp., decided to make Android operating system (originally developed out of Linux kernel) an OSS so that the complex problems in the system become solved by diverse community of users. Felin and Zenger (2014) hold that firms tend to interact with innovation communities in highly complex ways as they seek to generate value and look for optimal ways to organize their problem solving and innovation.

These studies show that, first, openness (having porous organizational boundaries) is advantageous. Firms that opt for openness can use external knowledge to complement internal knowledge and fill the existing knowledge gaps (Lichtenthaler & Lichtenthaler, 2009; Chesbrough, 2006c). They can also access scarce financial and knowledge resources (Drechsler & Natter, 2012) to enhance their R&D and innovative performance (Fey & Birkinshaw, 2005; Laursen & Salter, 2006b); to reduce costs and risks of R&D, and to get involved early in new technologies and business opportunities (Vanhaverbeke, van de Vrande & Chesbrough, 2008). Second, the decision to open up and remain porous are not made blindly; meaning that a firm's innovation strategy, the capability to manage and integrate the external knowledge resources, governance mechanisms, and its ability to appropriate the value as created collaboratively drive its openness strategy (Drechsler & Natter, 2012). Third, the absence of the necessary absorptive capacity (Cohen & Levinthal, 1990),

appropriate governance structure and appropriability mechanisms, can cast doubts on a fruitful openness process.

However, “Too much openness can negatively impact companies’ long-term innovation success, because it could lead to loss of control and core competences” (Enkel, Gassmann, & Chesbrough, 2009, p. 312). It is true that taking a closed approach to innovation “does not serve the increasing demands of shorter innovation cycles and reduced time to market” (Enkel et al., 2009, p. 312), but one cannot approach openness without considering the context and its downstream consequences. For this reason, past research has emphasized that there is still a lack of a clear understanding of the underlying mechanisms of the degree of openness and that an appropriate balanced approach to OI is key to success (see, Dahlander & Gann, 2010; Enkel et al., 2009).

In summary, the review of the literature on openness informs us about three main realities. First, openness is the *essential condition* for adopting OIP. This simply means that if a firm does not adopt an open approach to its innovation processes, it cannot benefit from the advantages discussed in OI literature. Second, openness is *not the sufficient condition* of OIP. For instance, by merely seeking and integrating ideas, talents, IPs, and other enablers that exist beyond firm boundaries, a firm cannot necessarily ensure firm-level innovation success and stronger or strategic market positioning. Thus, issues like appropriate governance, managing network externalities, honing the right absorptive capacities, to name a few, are among the significant factors that must be discussed at a higher organizational level. Third, and the main *caveat* in the openness literature, is the overemphasis of the existing literature in framing openness as dominantly a firm-centric phenomenon. Put simply, the decision to become more open at different phases of the innovation process has become subject to firm-level managerial discussions. Such firm-based or dominantly firm-centric (see, Piller & West, 2014, p. 29), and limited conceptualization of

openness tend to inhibit our understanding from broader implications of the openness in the open and distributed innovation communities and innovation ecosystems. For instance, in case of OSS industry where there is a heterogeneous and voluminous involvement of different individuals as well as organizations from private and public sectors, high-tech and low-tech manufacturing or services industries, the nature of openness and the way it is governed and managed may be different. For this reason, OSS industry presents a special case for studying openness in a more detailed manner— an investigation that also expands the boundaries of this concept within OI literature.

## 2.4 The theoretical perspectives on open innovation

### 2.4.1 Open innovation: Modes, models, and mechanisms

Chesbrough (2006b, p. 1) defines OI as “the use of purposive inflows and outflows of knowledge to accelerate internal innovation, and expand the markets for external use of innovation, respectively”. This conceptualization of OI forms the basis for the majority of research development in the field. However, most recently, Chesbrough and Bogers (2014, p. 4) build upon this earlier perception of OI and claim that OI “ought to be conceptualized as a *distributed* [emphasis added] innovation process that involves purposively managed knowledge flows across the organizational boundary”. The recent conceptualization, according to Chesbrough and Bogers (2014), connects the OI concept closer to its types: Outside-In (inbound), Inside-Out (outbound), and Coupled OI, as well as to its associated mechanisms: pecuniary and non-pecuniary flows.

According to Chesbrough and Bogers (2014, p. 16), “purposive inflows and outflows of knowledge” connects the OI practice with the spillovers’ literature. Originally,

firms cannot be fully in control of the results of their R&D investments and therefore spillovers are, and can be perceived as a cost to the focal firm as it is difficult to completely avoid or manage them (see, Kenneth Arrow, 1962). However, OIP is distinguished from this view of wasteful R&D investment in that “in the open innovation model of R&D, spillovers are transformed into inflows and outflows of knowledge *that can be purposively managed*” (Chesbrough & Bogers, 2014, p. 17). Accordingly, following OIP, firms need to design “specific mechanisms” to channel inflows and outflows of knowledge. Thus, taking into considerations spillovers’ concept and particular mechanisms to harness OI-based R&D investments, Chesbrough and Bogers (2014, p. 17) refine the OI concept by defining it as:

“A distributed innovation process based on purposively managed knowledge flows across organizational boundaries, using pecuniary and non-pecuniary mechanisms in line with the organization’s business model.”

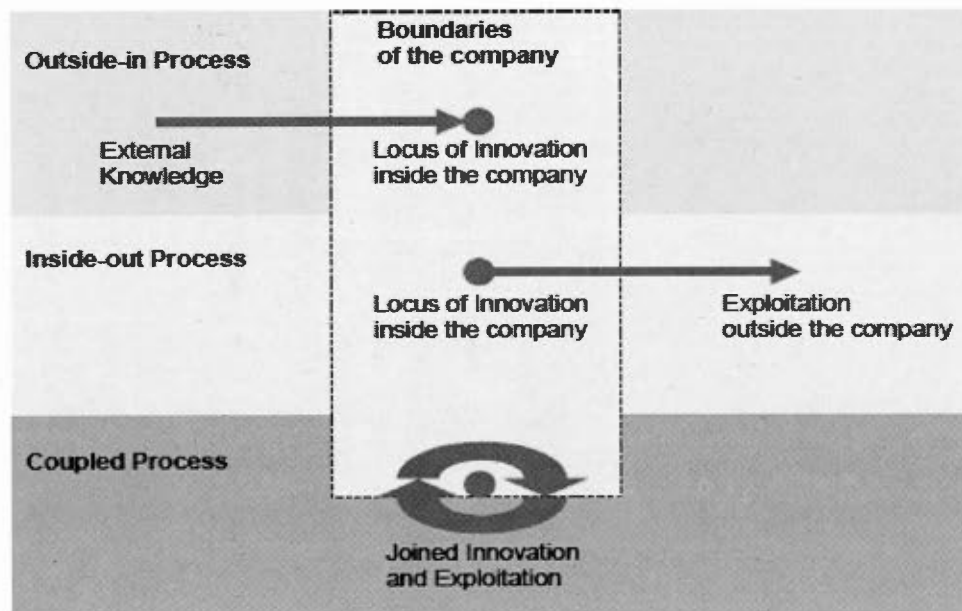
The purposeful management of knowledge spillovers also sheds light on two major modes of OI based on the direction of knowledge flows across firm boundaries. These two are: 1) *Outside-in* (or inbound); and 2) *Inside-out* (or outbound) OI (Chesbrough, 2003a; Chesbrough & Bogers, 2014). The ‘outside-in’ mode of OI concerns the inflow of knowledge to the focal firm where emphasis is on tapping into external sources of knowledge and resources by deploying internal processes. The ‘inside-out’ mode of OI, on the other hand, involves outflow of knowledge from the focal firm where emphasis is on commercializing the internal knowledge, for example, in form of untapped latent patents and inventions (see, for example, Chesbrough 2003a).

Gassmann and Enkel (2004, p. 1) define the two modes of OI in the following manners. The outside-in process refers to “enriching a company’s own knowledge base through the integration of suppliers, customers, and external knowledge sourcing can increase a company’s innovativeness”, while the inside-out process concerns “the

external exploitation of ideas in different markets, selling IP and multiplying technology by channeling ideas to the external environment”.

In addition to these two modes (or types) of OI, Gassmann and Enkel (2004; Enkel et al., 2009) further identified a third mode based on an empirical analysis of 124 firms. The third mode of OI: “*Coupled Process*” refers to “linking outside-in and inside-out by working in alliances with complementary companies during which give and take are crucial for success” (Gassmann & Enkel, 2004, p. 1). Figure 2.1 shows the three modes of OI based on Gassmann and Enkel (2004)’s approach to OI.

Figure 2.1. De-Coupling the Locus of Innovation Process



Source: Adopted from Gassmann and Enkel (2004, p. 6)

Further, according to Gassmann and Enkel (2004)’s empirical data, a firm does not necessarily use all three processes; rather, each firm opts for one primary process, yet it integrates some elements of other two. Additionally, Gassmann and Enkel (2004) mention that firms that follow coupled-processes approach are required to co-operate



with others in strategic networks where give and take of knowledge lies at the center. Following the classification made by Gassmann and Enkel (2004), in my opinion, the most challenging approach to openness is coupled-processes as they require a firm to create and juggle a combination of different set of capabilities involved with both processes.

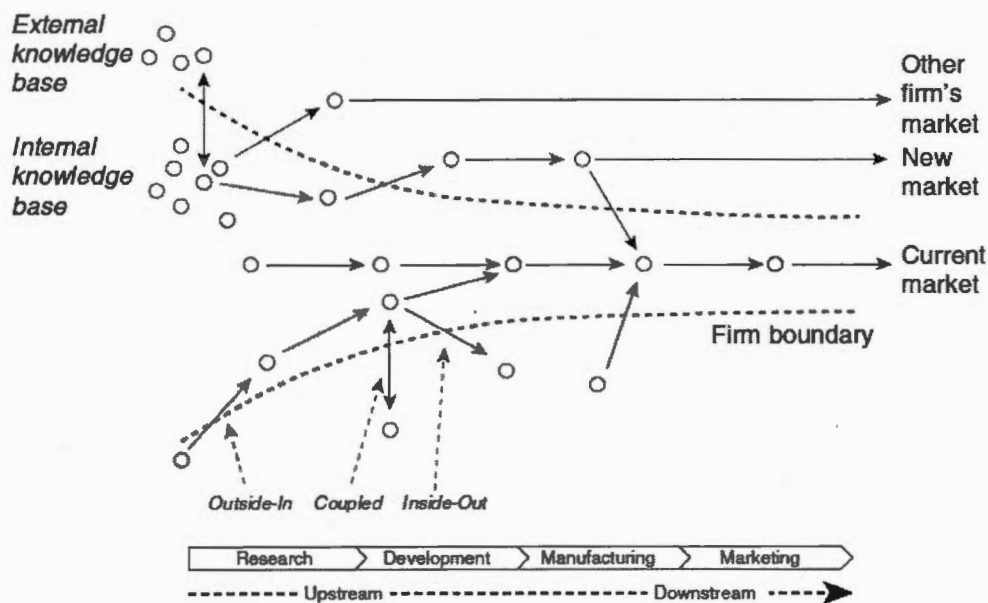
Gassmann and Enkel (2004) further specify different capabilities for each process. For example, absorptive capability is required for engaging a firm in the outside-in process while multiplicative capability (i.e. a firm's capability to multiply and transfer its knowledge to the outside environment) is needed to execute inside-out process. Further, multiplicative capability is a complex capability as it is tightly interrelated to a firm's knowledge transfer capability and the capability to select appropriate partners. Relational capability, on the other hand, is one used for coupled processes and it draws on the idea that a firm's value is strongly related to its capability to build and maintain relationships with partners to enable joint development in strategic alliances (Dyer & Singh, 1998).

More recently, Dahlander and Gann (2010) classify OI approaches into four categories: 1) *outbound innovation (non-pecuniary)*; 2) *outbound innovation (pecuniary)*; 3) *inbound innovation (non-pecuniary)*; and 4) *inbound innovation (pecuniary)*. The authors also discuss advantages and disadvantages of each class of openness. For instance, firms that reveal their internal resources *without* expecting *immediate* financial rewards; yet, seeking indirect benefits to the focal firm (e.g., OSS companies) fall under the first category.

In the recent book published on OI: "New Frontiers in Open Innovation" (edited by Henry W. Chesbrough, Wim Vanhaverbeke, Joel West, 2014), Chesbrough and Bogers (2014) officially incorporate the third mode of OI into the original conception and demonstrate the three through the classical innovation funnel (see Figure 2.2).

Figure 2.2, demonstrates the inflow and outflow paths of knowledge across firm boundaries under OI regime. It further highlights the connection between upstream R&D to manufacturing and marketing and emphasizes downstream activities in the overall innovation process (Bogers & Lhuillery, 2011). Chesbrough and Bogers (2014) claim that this overall model emphasizes “the importance of considering all activities from invention to commercialization in order to create *and* capture value from ideas and technologies” (see also, Chesbrough, 2006a; West & Bogers, 2014).

Figure 2.2. The Open Innovation Model



Source: Adopted from Chesbrough and Bogers (2014, p. 18)

Despite the efforts of pioneering authors to conceptualize OIP through the three ‘core’ processes (the inbound, the outbound, and the coupled), majority of research on OI, so far, has been focused on ‘outside-in’ OI processes. This narrow focus has led to dearth of empirical evidence on the ‘inside-out’ and the ‘coupled’ modes of OI

(Chesbrough & Bogers, 2014). For instance, out of 165 papers on OI, only 118 articles are concerned with inbound OI, 50 of the reviewed papers focused on outbound OI, and 70 of the sampled articles consider coupled mode of OI (West & Bogers, 2014, p. 818).

The review of three OI processes draws our attention to the dearth of understanding of “the interactive and reciprocal nature of such coupled innovation processes” (Chesbrough & Bogers, 2014, p. 19. More specifically, in case of technological collaborations embedded in OSS projects, investigating the coupled processes gain significance. This is mainly because OSSTCs include multi-sided interactions among a plethora of firms, community of developers and freelance individuals who collectively develop software products and modules in the commons.

#### 2.4.2 Overlap between user innovation and open innovation perspectives

User Innovation (UI) concept (see, von Hippel, 1988, 2005, 2010), like OI, provides an alternative perspective to closed innovation model. UI places ‘users’ on the driver’s seat of the innovation process, viewing them not as mere consumers but as empowered “self-manufacturers” who can create their own products and services out of necessity or for fun; individually or in group(s) (e.g., collaborating through communities which includes firms and individuals). At the heart of UI’s perspective lies three chief assumptions: 1) users have unique “sticky” information about their needs; 2) when enabled, they will create solutions to those needs; and 3) they may freely reveal their results to others (see, von Hippel, 2010). Recently, Piller and West (2014, p. 33) have mentioned that UI’s literature has grown from being primarily focused on “innovating users” and has further developed into a conception which involves “interaction among users and firms”. However, there is dearth of research investigating “in-depth the process of collaboration between users and firms”.

UI, despite being different from OI, shares certain traits with OI. Most important of all, when it comes to realizing that the knowledge necessary for innovation is widely dispersed beyond the conventional boundaries of the firm, the two current approaches to innovation; namely, OI (Chesbrough, 2006b) and UI (von Hippel, 2005) converge (for detailed discussion see, Bogers & West, 2012). Additionally, both concepts are in sharp contrast with the traditional vertically integrated innovation framework (see, Bogers & West, 2012, p. 64).

However, despite having points of commonalities and overlap, the two approaches diverge because they tend to study “different phenomena” (Piller & West, 2014, p. 29). While OI is a rather “firm-centric paradigm that is primarily concerned with leveraging external knowledge to improve internal innovation and thus the firm’s economic performance”; UI, on the other hand, “is mainly about individuals using innovation to address their own (often unique) needs, without regard to firm success and often as part of a socially embedded community”.

In short, three main distinctions exist between UI and OI (Piller & West, 2014; see also, Bogers & West, 2012). *Firstly*, OI success in private sector depends heavily on strong appropriability and aggressive IP enforcement ( e.g., Chesbrough, 2003b; West, 2006) while UI paradigm mainly favors “free revealing” as in giving up the appropriability voluntarily in order to promote innovations (Harhoff, Henkel, & von Hippel, 2003; Henkel, 2006). *Secondly*, to secure economic incentive for the innovating firm, the private control model is implicitly built into OI studies; while the expansion of UI model mostly favors collaboration among actors which also involves sharing of the benefits through collective or private-collective innovation models (see the major works on this front by von Hippel & von Krogh, 2003; West, 2003).

Thirdly, the distinction between ‘money markets’<sup>11</sup> and ‘social markets’<sup>12</sup> also influence the organization of individuals’ participation in the innovation process (Piller, Vossen, & Ihl, 2012). For example, those individuals who interact in social markets tend to expend more effort in exchange for no payment than they would expend when they receive low payment in monetary markets (Heyman & Ariely, 2004).

Thus, Piller and West (2014, p. 36) claim that these distinctions between OI and UI can shed light on “areas of tensions between the interests of firms and those of individual users when they collaborate”. This reflects the contrast between personal utility vs. private economic returns and the resulting tensions if they are not reconciled.

In order to extend the research on coupled processes, Piller and West (2014) identify four important ‘*dimensions*’ of them along with proposing a ‘four-phase model’ of interactive coupled OI and how firms can manage such model with individual users. Such model integrates the previous understanding on ‘inbound’ OI with collaborative innovation (including collaborative tools and processes).

The four dimensions of coupled OI mode, as identified by Piller and West (2014, p. 38) are: 1) *external actor*; 2) *coupling typology*; 3) *impetus for collaboration*; and 4) *locus of innovation*.

The ‘nature of the external actor’ concerns firms, and individuals as well as non-for profit organizations. This dimension therefore extends the earlier conception by

---

<sup>11</sup> Money markets are markets for external innovation and are organized around economic (monetary) incentives exchanged for ideas and solutions (e.g., Terwisch & Xu, 2008; Jeppesen & Lakhani, 2010; Boudreau et al., 2011).

<sup>12</sup> Social markets are formed based on social exchange relations and are mainly built upon the non-monetary incentives for participants such as enjoyment or task achievement (von Hippel & von Krogh, 2003, 2006), for outcome expectations that enhance their own use experience or that of others (Harhoff et al., 2003), or through norms of mutual cooperation and reciprocity (Lakhani & von Hippel, 2003).

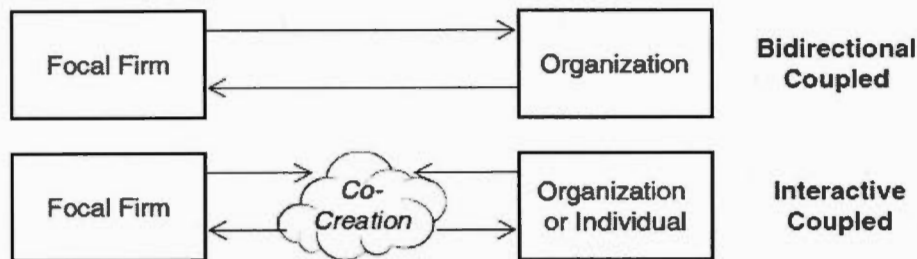
including non-for profit organizations and individuals and consequently raising the issues of incentives, coordination and governance of the collaboration between two parties. Next is the 'typology of relationships' with the external actors. This proposed typology includes dyadic (single partner), network (multiple partners) and community (a new inter-organizational entity) relationships. Further, the 'impetus for collaboration' distinguishes between top-down and bottom-up modes of initiating, directing and implementing collaboration. Lastly, 'locus of the innovation process' is concerned with two approaches: a) bidirectional; and b) interactive.

Bidirectional approach to innovation captures the case in which two actors, which are typically two organizations, independently engage in the innovation activity, yet they also share useful knowledge. Piller and West (2014) view bidirectional approach a concept that closely describes what Gassmann and Enkel (2004) mean by combining inbound and outbound flows of knowledge, i.e. coupled process of OI. On a more extreme level, interactive approach captures a knowledge creation which takes place outside firm boundaries. It differs from bidirectional approach in that the locus of innovation is not within collaborating firm; it also involves issues like how the joint knowledge creation process is governed, how the returns on investments are to be appropriated, and so on (see also Chesbrough, 2011).

The contribution of Piller and West (2014) to build upon the couple OI process is through breaking down the coupled mode of OI into two and emphasizing the coupled process as "*interactive, collaborative process of joint value creation*". Figure 2.3 depicts the authors' model of coupled mode of OI.

Moving on towards building a 'four-phase process model' of interactive coupled OI between firms and users, Piller and West (2014, pp. 40-46) propose four major steps of the process model: 1) *defining*; 2) *finding participants*; 3) *collaborating*; and 4) *leveraging*. The third phase is most relevant to the present research.

Figure 2.3. Two Forms of Coupled Open Innovation



Source: Adopted from Piller and West (2014, p. 39)

Collaboration involves the process of co-creating the knowledge which does not necessarily exist within a single partner. As claimed by Piller and West (2014, p. 44), “research on the joint creation process of such knowledge [external knowledge] has been comparatively rare in the open innovation literature”. In fact, there is dearth of empirical research investigating the “structures and processes supporting collaborative knowledge creation with external actors” (Piller & West, 2014, p. 44) (see, e.g., Blazevic & Lievens, 2008). Further to this dearth of research, Piller & West (2014, p. 45) mention that even “the original UI literature on lead users did not look upon the collaboration stage, except ... for collaborations within communities of innovating users” (e.g., Franke & Shah 2003; von Krogh, Spaeth, & Lakhani, 2003). Additionally, these studies have not paid attention to collaboration between users and firms (Piller & West, 2014, p. 45). Therefore, Piller and West (2014, p. 45) classify the research gaps on the “collaboration stage of coupled OI” into three groups. These are: 1) *governance of collaboration process* (i.e., organizing, monitoring and policing collaborations); 2) *tools and collaboration infrastructures facilitating governance* (i.e., software tools); and 3) *internal attitudes and capabilities of focal firm supporting the collaboration* (i.e., using the services of specialized intermediaries and brokers for OI).

To summarize, first, OI is a rather “firm-centric paradigm” while UI “is mainly about individuals” who do not concern a firm’s profitability and economic success. Second, success of OI heavily depends on strong appropriability and aggressive IP enforcement while that of UI rests upon free revealing to increase follow-up adoptions and expand network externalities. Third, as a firm-centric approach, OI favors private control models while UI favors collective innovation model and at times a mix of private and collective development. Fourth, the target market of OI adopted by private firms is money markets, while that of UI is mainly social markets. The distinction between market types influences individuals’ incentives for their contributions and their organization.

In short, based on the above discussions, we can conclude that both OI and UI suffer from shortcomings that make them inadequate to explain intricacies of OSSTCs. Perhaps it is for these reasons that Piller and West (2014, p. 47) highlight the need for empirical research to clarify “hybrid models” of innovation where UI and OI are combined. OSSTCs is such case as they include close collaboration among individuals (usually free software developers who write codes under GNU GPL licensing) and professional IT firms (SMEs) as well as large software corporations (e.g., Microsoft, Google, and IBM) that use these software tools, add to them and improve on them before integrating them into their proprietary products and finished goods. By studying OSSTCs, we can bridge both literatures (OI and UI) and extend our understanding of coupled and interactive shared innovation processes.

#### 2.4.3 Implications of private-collective action model for open innovation paradigm

OI perspective can also theoretically benefit from conceptual developments made on the front of private (Demsetz, 1967) and collective (Olson, 1967) innovation models



as well as the (fairly) recent mixed approach proposed by von Hippel and von Krogh (2003).

There are two major models of innovation: 1) *Private investment model*; and 2) *Collective action model*. 'Private investment model' is based on the premise that innovation will be supported by private investment, and that private returns can be appropriated from investments made (Demsetz, 1967). That is why "property rights are an instrument of society and derive their significance from the fact that they help a man form those expectations which he can reasonably hold in his dealings with others" (Demsetz, 1967, p. 347). Following Demsetz (1967)'s and several others' (Arrow, 1962; Dam, 1995; as cited in von Hippel & von Krogh, 2003) line of reasoning, by granting innovators some limited rights to their innovations, a society encourages private investment in innovations. This reflects Demsetz (1967, p. 348)'s thought on the primary function of property right as "guiding incentives to achieve a greater internalization of externalities"; therefore, "any free revealing or uncompensated 'spillover' of proprietary knowledge developed by private investment will reduce the innovator's profits from its investment" (von Hippel & von Krogh, 2003, p. 213). However, a society adopting a purely private investment model will incur a certain loss as property rights block a portion of society to access innovations. However, von Hippel and von Krogh (2003) believe that society voluntarily suffers from this social loss in the hope that innovators have enough incentives to make further investments to create novel knowledge.

Contrary to private investment model is the 'collective action model' pioneered by Olson (1967). This model applies to the provision of 'public good' (also known as a common or collective good) defined as "any good such that if any person  $X_i$  in a group  $X_1, \dots, X_2, \dots, X_n$  consumes it; it cannot feasibly be withheld from the others in that group" (Olson, 1967, p.14). In fact, under conditions of market failure, innovators tend to collaborate to produce a public good (von Hippel & von Krogh,

2003). The logic governing the collective action model, therefore, requires that those who have invested in an innovation abandon their right and control of it thereby turning that into a public good. This approach to innovation avoids society to incur social loss that is indubitably imposed by the private investment model; yet, it causes another problem, i.e., it takes the incentive to innovate away from innovators by not letting them appropriate directly from their investments. This problem manifests itself in several forms one of which is the “free-riding” problem in collective action (see Hargrave & van de Ven, 2006; von Hippel & von Krogh, 2003; Bonaccorsi & Rossi, 2003).

In context of OSS technology, individuals have the choice to follow more or less both models by choosing a more liberal or more limiting licensing mode. However, in principal, one can access, and study any OSS’ source code no matter which licensing type is applied to it. The flexibility that leans more towards collective action model has intrigued several scholars.

For example, von Hippel and von Krogh (2003) developed a mix of the two models based on their empirical observation of OSS projects. The authors maintain that the two models have ignored a middle ground solution that is a happy marriage between private investment and collective action. They call this third option “*private-collective*” innovation model (PCIM) (“best of the both worlds”) which offers enough incentives for both models to coexist.

To begin with, PCIM eliminates the assumption in private investment model which asserts that free revealing of innovations developed by private investors (innovators or code developers) will represent a loss of private profit. To support this argument, von Hippel and von Krogh (2003) draw on several issues<sup>13</sup> and conclude that under

---

<sup>13</sup> Firstly, private investment model ignores individual user-innovators which seek (direct internal-use benefits at the expense of placing too much emphasis on manufacturers as innovators who are looking for direct private commercial market benefits (e.g., von Hippel, 1988; von Hippel & von Krogh, 2003).

some conditions (which hold for OSS projects) free revealing may actually result in a net gain in profit for the innovators. For example, as free revealing enhances the diffusion of OSS there is good chance that innovators will benefit from the profits associated with their innovation through network effects.

Secondly, the PCIM defies the collective-action-model-based assumption which holds that a free rider will enjoy benefits from the completed public good just the same way contributors would enjoy those benefits. This deviation is based on the observation that there are certain private benefits associated with OSS projects as public goods which are not open to free riders. This argument implies that OSS projects are more *process oriented* (see also, Bonaccorsi & Rossi, 2003) than product oriented. For this reason, a free rider's benefits are rather limited vis-à-vis an active user-innovator/contributors. For example, the problem-solving process involved with OSS projects' solutions development entails several benefits such as: technical learning opportunities and enjoyment (Kohanski, 1998; Hermann et al., 2000); sense of ownership and control over contributor's work product (Lakhani & von Hippel, 2000); the freedom of choice to work on what type of projects and taking a preferred technical approach, therefore, respecting self-interests of contributors (von Hippel & von Krogh, 2003). Similarly, Artur (1997) notes that the previous coding and the learning obtained from its process can increase the user's returns on learning in the future activities. Therefore, these benefits are considered as private benefits that are not obvious in typical collective action model of innovation.

Furthermore, von Hippel and von Krogh (2003) describe a situation which concerns social integration of OS code developers. As these user/developers' incentives may

---

Secondly, private investment model does not find free revealing of novel codes sensible unless costs of free revealing are less than its benefits. Well, as there is low level of rivalry between some potential contributors and adopters (e.g., Lakhani & Wolf, 2005), innovators incur low costs which are associated with freely revealing and diffusing the codes. This means that low level of rewards looks appealing too (von Hippel & von Krogh, 2003). These rewards can be elevated reputations, expected reciprocity, and incentives to build a community (Lerner & Tirole, 2000).

change over time, their learning rewards may become exhausted, and the value of the social category (e.g., listed and perceived as core-developer status), they have been assigned to, start depreciating. Therefore, it is likely that the developers' level of collaboration decreases.

Additionally, PCIM potentially invites further research concerning the "role and nature of leadership" in relation to sustaining the ongoing activity in OSS projects and communities. For example, some authors believe that as the OSS group gets larger and larger they need to establish a central authority or leadership to monitor the group members and impose sanctions on free-riders become more obvious (Hardin, 1982; Swanson, 1992). The importance of leadership role in developing OSS projects has been already raised (Pavlicek, 2000); yet, the nature of leadership seems to have remained as an elusive concept. For example, one of the norms of hacker community is that the to-be-done tasks cannot be forced by the leader (Himanen et al., 2001; Raymond, 1999); yet the intricacies as how the job is done have not been illuminated. Therefore, it is important to gain more insights on the nature of leadership in OSS development process as representative of OI practice.

Overall, the case of OSSTC is one that follows PCIM. However, it is a very dynamic case because contributors' incentives change over time and thus, in the absence of strong legal and contractual bindings, the community leadership needs to adapt its strategies to align the interests and create new ones in order to keep the project on the right track. The significance of these interplays makes the study of OSSTC interesting.

## 2.5 Useful contributions of open innovation to open source software

### 2.5.1 What open innovation teaches us about open strategy

Based on observation of firms that have successfully adopted OI practices, Vanhaverbeke and Cloudt (2006, p. 260) posit that “open innovation practices have to be embedded in firm’s strategy”. In this section, I touch upon “Open Strategy” (Appleyard & Chesbrough, 2016; Chesbrough & Appleyard, 2007) and ‘Business Model’ (BM), two main topics that link OI concept to strategy. Further, I review the empirical works conducted on the connection between OI and strategy– paying more attention to the relationship between BM and OI mainly because “business model thinking as part of firm’s strategy is at the heart of open innovation” (Vanhaverbeke & Cloudt, 2006, p. 260).

### 2.5.2 Open strategy

The field of “open strategy” is being shaped in recent years (Appleyard & Chesbrough, 2016; Chesbrough & Appleyard, 2007). Here, I briefly define strategy and then move towards the open strategy concept.

Chandler (1962, p. 13) defines strategy as “... *the determination of the basic long-term goals and objectives of an enterprise, and the adoption of courses of action and the allocation of resources necessary for the carrying out of these goals*”. In the field of *business and management literature*, Evered (1983) recommends Andrews (1980)’s definition of ‘corporate strategy’ as the best fitting one. He defines strategy as:

“the pattern of decisions in a company that determines and reveals its objectives, purposes or goals, produces the principal policies and plans for achieving those goals, and defines the range of business the company is to

pursue, the kind of economic and human organization it is or intends to be, and the nature of the economic and noneconomic contribution it intends to make to its shareholders, employees, customers and communities.” (Andrews, 1980, p. 18)

In context of OI, the term “open strategy” is used to link strategy with OIP. According to Chesbrough and Appleyard (2007, p. 58), “if we are to make strategic sense of innovation communities, ecosystems, networks, and their implications for competitive advantage, we propose that a new approach to strategy – open strategy – is needed”.

To justify open strategy, Chesbrough and Appleyard (2007) claim that “*traditional business strategy* has guided firms to develop defensible positions against the forces of competition and power in the value chain, rather than promoting openness” (Chesbrough & Appleyard, 2007, p. 57). In fact, they believe, “all traditional views [of strategy] are based upon *ownership* and *control* as the key levers in achieving strategic success” (Chesbrough & Appleyard, 2007, p. 60; emphasis added). The focus of a firm’s strategic management, following the traditional view, has been largely on what is going on within the firm; i.e., internal value chain to the exclusion of the potential value of external resources (e.g., innovation communities) not owned by the firm.

Although “open strategy” sounds like a revolutionary topic in the IP dominant market economy, its mention rings the bell of the ideas put forward by Roger Evered in 1983. Evered (1983) discusses the concept of strategy from a historical/military, business management as well as futures research perspectives in order to argue that the corporate conception of strategy is rather a mix of military conception of strategy and that of the futures research. He mentions that strategy is “....essentially that of collectively learning [on] how to transform our collective selves” (Evered, 1983, p.

71). As so argued, key attributes such as *Trust, openness, collaboration, organizational learning, error embracing* and *sharing* which are unique to futures research become part of corporate strategy which mediates the other two modes of strategy. In view of Evered (1983)'s, perhaps, open strategy is a good start to expand the concept of traditional approach to strategic thinking in strategic management field.

The core concept of open strategy; i.e., making use of external resources, innovate collectively and gain competitive edge can also be captured in examples put forth in Eric von Hippel's '*Democratizing Innovation*' (2005, p. 14). For example, the evolving pattern of the locus of product development in kite surfing illustrates how users can displace manufacturers from the role of product developer. In this industry, the collective product-design and testing efforts of a user innovation community has clearly become superior in both quality and quantity relative to in-house development [R&D] efforts that manufacturers of kite surfing equipment can justify. Accordingly, manufacturers of such equipment are increasingly shifting away from investing in brand-new product design by more focusing on product designs primarily developed and tested by user innovation communities.

Therefore, Chesbrough and Appleyard (2007) recognize '*emerging anomalies*' (e.g., OI products such as Linux operating system, MySpace, Wikipedia, as well as (user) innovation communities, etc.) and view them as the reasons which have shaken the basic tenets of traditional business strategy. They see these anomalies as agents of change which brought about the shift in our strategic focus *from ownership to the concept of openness* which in turn requires us to rethink the processes that underpin value creation and value capture. Open strategy, in such sense, balances the tenets of traditional business strategy with the promise of OI through use OI-based BMs and deployment of community driven innovations. (Chesbrough & Appleyard, 2007). Duarte and Sarkar (2011, p. 455, Emphasis added), viewing OI strategy positively,

conclude that “open innovation strategies, whether formal or not, intend to increase the *competitiveness* of firms, in particular the speed with which they introduce innovations in the market”.

Here, we can argue that several technological anomalies render the tenets of Porter’s Five Forces (Porter, 1980a), which are the basis for gaining competitive edge, counterintuitive. For example, “Microsoft’ masterful cultivation of the Five Forces of Porter has done little to slow Google’s meteoric rise in market capitalization” (Chesbrough & Appleyard, 2007, p. 61). Another example would be the magic of Linux operating system that has defied the hazard of ‘the tragedy of commons’ by empirically showing us that the more people work on and modify the product the more value is created. In fact, the value of the product increases by every consumer added to the system and the innovation process becomes that of a self-sustained one.

Although one perceives open strategy a promising highroad in many sectors, especially in view of the past success stories, one cannot ignore the perils of missing the *balance between value creation and value capture*. In fact, these two poles of the BM seesaw will never go out of fashion no matter what view of strategy one may take.

As such, Linux, which is believed to be “the poster child for open innovation” (Chesbrough & Appleyard, 2007, p. 62), has been institutionalized through the creation of Open Source Development Labs which is funded partly by IBM, Intel, HP and Oracle. This has been an attempt to reach a self-funding model by monetizing open source opportunities that complemented Linux. This re-structuring move to capture further value, if perceived by a substantial portion of the community of developers as far from their ideals of a community-based meritocracy, may lead to the collapse of the community of developers (Chesbrough & Appleyard, 2007). Therefore, in formulating an open strategy, how the value is created (open knowledge



creation and invention) and how the value is appropriated by actors remain elusive concepts which rely heavily on dynamics of the business ecosystem and accordingly constitute an area for further research.

Following this line of argument, one can observe the interplay between the tenets of the traditional business strategy with concepts of open strategy in the four categories (development, hybridization, complements, and self-service) of Open Source Software Business proposed by Chesbrough & Appleyard (2007). For example, adopting hybridization BM requires the firm to make proprietary investments which rely on intellectual property ownership for add-ons (proprietary extensions) (Chesbrough & Appleyard, 2007). This example highlights two issues. First, the open strategy differs from the traditional trend in that it draws on a firm's open BM in value creation. Second, open strategy approximates traditional approach by creating ownership to capture value.

Thus, following the widespread consensus among scholars that every firm has a BM, and it needs to be functioning to make the firm relatively successful (Magretta, 2002). In addition to knowing that a BM's chief role is to create and capture value (e.g., Zott et al., 2011; Chesbrough, 2007a; Brandenburger & Stuart, 1996, to name a few), the challenge remains with adoption of an open strategy and how a firm can successfully strike the right balance between the value creation and value capture. In my opinion, open and traditional business strategies are complementary, and each firm needs to mix up its own cocktail to benefit from its BM optimally.

### 2.5.3 The relationship between BM and OIP

There are two central themes to Chesbrough's concept of OI. The first concerns the *ideation* part, while the second reflects the *commercialization* one.<sup>14</sup> In fact, 'openness' can be applied to either stages or both simultaneously. For example, a firm may develop a novel idea (internally) but does not bring it to market by itself. Rather, the company opts for collaborating with or even selling the idea to another firm that has the capacity, resources and the expertise to commercialize it. Thus, to optimize or get the most out of OIP, firms *must open their BMs* by actively searching for and exploiting outside ideas and by allowing unused internal technologies to flow outside, where other firms can unlock their latent economic potential (Chesbrough, 2007b, p. 22). Following this argument, we may shift the spotlight on BM discussions and how BM relates to OIP.

As Teece (2010, p. 174) states, "the concept of a business model has no established theoretical grounding in economics and business studies". BMs neither have a place in economic theory, nor have an acceptable place in organizational and strategic studies (Teece, 2010). Teece (2010, p. 175) further cautiously comments on the lack of theoretical grounding of BM discussion in the field of economic theory by highlighting the possibility that such shortcoming may stem from "the ubiquity of theoretical constructs that have markets solving the problems that- in the real world- business models are created to solve".

On a more detailed note, Teece (2010, p. 175) explains that value creation and value capture are merely perceived as assumptions so much so that "*inventions are often assumed to create value naturally and, enjoying protection of iron-clad patents, firms can [therefore] capture value by simply selling output in established markets, which*

---

<sup>14</sup> The original invention needs to be brought to the market to be considered an innovation (Schumpeter, 1934).

*are assumed to exist for all products and inventions*". In so thinking, creating a value proposition for customer and crafting value appropriation mechanisms are not important as customers eventually buy as long as the price falls below the utility they gain with emphasis on competitive market prices (Teece, 2010). Yet, Teece (2010, p. 175) claims that "equilibrium and perfect competition are a caricature of the real world... customers don't just want products; they want solutions to their perceived needs". Following this argument, more attention will be given to developing theories that explain BM architecture and includes customers as part of BM strategic design.

In consideration of these remarks, therefore, two major issues arise. First, to understand OI business models (OIBMs) and make a theoretical contribution to its field of study, we need to gain a more insightful perspective on Business Model (BM) concept per se, and its components and then build a connection between BM concept and OIP; with OIBM being an indispensable part of it.

Second, in the digital and knowledge-based economy firms are enabled by Information and Communication Technologies (ICTs) and have had the potential to experiment novel forms of 'value creation mechanisms' which are networked in the sense that value is created in concert by a firm and a plethora of partners for multiple users (Zott et al., 2011). This reconceptualization of value has attracted the attention of management scholars, who have employed the concept of the BM in their attempts to explain value creation in networked markets (e.g., Zott & Amit, 2009). In fact, recent review of research on BM and strategy reveals "a strong consensus that BM revolves around customer-focused value creation" (Chesbrough & Rosenbloom, 2002; Mansfield & Fourie, 2004; as cited in Zott et al., 2011, p. 1031). Such perspective coupled with the doctrine of networked value creation process puts customers (and/or clients) under spotlight as playing a pivotal role in creating value and being the target for value creation. Similarly, Teece (2010, p. 176) emphasizes the connection between BMs and consumers by highlighting the role of BM as

“necessary features of market economies where there is consumer choice, transaction costs, and heterogeneity amongst consumers and producers, and competition”. Therefore, the degree to which BMs are adapted to customer needs and their business environments is among distinguishing features of different BMs where “a good business model will provide considerable value to the customer and collect... a viable portion of this in revenues” (Teece, 2010, p. 179).

#### 2.5.4 Business model: Concepts, definitions, and perspectives

Every firm has a Business Model (BM) even if it is not explicitly articulated (Teece, 2010; Chesbrough, 2007a, p.12). It is a concept often used in management practice and theory; yet BM is rather an opaque notion which lacks conceptual clarity and definition uniformity (Zott, Amit, & Massa, 2011; Morris et al., 2005; Chesbrough & Rosenbloom, 2002). As a case in point, Morris et al. (2005, p. 726) mention, “no generally accepted definition of the term ‘business model’ has emerged”. Similarly, others have claimed that despite the term’s ubiquitous usage, “it is seldom defined explicitly” (Chesbrough & Rosenbloom, 2002, p. 532). Even more recently, Zott et al. (2011, p. 1022) have mentioned that “surprisingly, however, the business model is often studied without an explicit definition of the concept”.

On a conceptual level, several authors (Zott et al., 2011; Morris et al., 2005) have highlighted the absence of BM’s definitional clarity as a source of challenge. This seems to obstruct further clarifying the nature and building components of BM. Definitional inconsistency may also create terminology confusion to the extent that notions like “business model, strategy, business concept, revenue model, and economic model are often used interchangeably” (Morris et al., 2005, p. 726). Zott et al. (2011, p. 1023) claim that the confusion due to lack of definitional clarity can also promote dispersion of perspectives and prevent cumulative progress on BMs.

Therefore, conceptual fuzziness has provided authors (e.g. Zott et al., 2011; Morris et al., 2005) with the motivation to conduct literature reviews to propose more comprehensive and unanimous definitions and develop frameworks to further study the BM concept as a unit of analysis. Among the prominent ones are Zott et al. (2011), Morris et al. (2005)'s, Teece (2010)'s, Chesbrough & Rosenbloom (2002)'s; to name a few. Table 2.2 presents a tabular review of most common definitions of BM concept compiled by the author.

Morris et al. (2005, p. 726) build upon the assumption that BM can be used as a “unifying unit of analysis” which in turn plays a facilitating role in entrepreneurship theory development. The authors conduct a literature review analysis to synthesize the current understanding of what BM is and what its components are. Based on their principal emphasis, Morris et al. (2005, pp. 726-727) classify the definitions into three levels: *economic*, *operational* and *strategic*. The first level concerns a firm's economic model so the logic of profit generation takes precedent. At this level, issues such as revenue sources, pricing methodologies, cost structures, margins and expected volumes are among the most relevant decision variables a firm is faced with. The second level refers to ‘architectural configuration’ where issues like internal processes and design of infrastructure become to forefront of value creation processes. Therefore, operational decisions regarding production/service methods, resource mobility, logistics and knowledge management play pivotal role. The third and the last is the strategic level. This level encompasses overall direction of the firm regarding “market positioning, interaction across organizational boundaries, and growth opportunities”.

The main goal of emphasizing strategic aspect of BM is to gain competitive advantage and sustainability. Thus, decision making stresses variables such as “stakeholder” identification, value creation, differentiation, vision, values, and

networks and alliances. Considering the three categories of BM and their points of emphasis, Morris et al. (2005, p. 727) introduce an integrative definition as follows:

*“A business model is a concise representation of how an interrelated set of decision variables in the areas of venture strategy, architecture, and economics are addressed to create sustainable competitive advantage in defined markets.”*

Furthermore, Morris et al. (2005) develop a six-component standard framework for characterizing a BM irrespective of venture type. Such framework includes three specific levels of decision making: *foundation*, *proprietary* and *rules* levels. The first involves *generic decision making* regarding what the business is all about; the second concerns development of *unique combinations among decision variables* which give the venture its marketplace advantage; and the third includes *guiding principles* which govern execution of decisions made at the prior two levels to make sure model effectiveness and usefulness (See Table 2.3).

Morris et al. (2005, p.734)’s study, as claimed, has several implications. First, the proposed framework (Table 2.3) offers the possibility to “design, describe, categorize, critique, and analyze a business model for any type of company”. Next, it allows researchers and practitioners to compare different BMs based on the levels and components. Third, it facilitates further empirical studies in the field of BM in the areas such as “the creation of general model taxonomies and investigations of relationships among the foundation level variables to modeling relationships between the model and a host of endogenous and exogenous variables”. In addition, this framework can be used to systematically assess BM validity proposed by entrepreneurs.

In another literature review-based study, Zott et al. (2011) show that the concept of BM has been used to explain three main phenomena: *e-business* and the use of information technology in organizations; *strategic concepts* like value creation, competitive advantage and firm performance; and *innovation and technology management*.

Table 2.2. Business Model Definitions

Author(s)	Year	Definition
Baden-Fuller and Haefliger	2013	"A system that solves the problem of identifying who is (or are) the customer(s), engaging with their needs, delivering satisfaction, and monetizing the value." (p. 419)
Aspara, Lamberg, Laukia, and Tikkanen	2013	"the corporate business model resides primarily in the minds of the corporation's top managers or top management team (TMT) members – essentially, it is <i>the corporate top managers' perceived logic of how value is created by the corporation, especially regarding the value-creating links between the corporation's portfolio of businesses.</i> " (p. 460)
Bucherer, Eisert and Gassmann	2012	"The business model abstracts the complexity of a company by reducing it to its core elements and their interrelations and thus specifies the core business logic of the firm. A business model is not static but must be managed and developed over time." (p. 184)
Demil and Lecocq	2010	"The business model concept generally refers to the articulation between different areas of a firm's activity designed to produce a proposition of value to customers." (p. 227)
Teece	2010	"The essence of a business model is in defining the manner by which the enterprise delivers value to customers, entices customers to pay for value, and converts those payments to profit" (p. 172). "A business model articulates the logic and provides data and other evidence that demonstrates how a business creates and delivers value to customers. It also outlines the architecture of revenues, costs, and profits associated with the business enterprise delivering that value." (p. 173)
Teece	2007	The business model reflects "the management hypothesis about what customers want and how an enterprise can best meet those needs, and get paid for doing so." (p.1329)
Seelos and Mair	2007	Business model is conceptualized as a "set of capabilities that is configured to enable value creation consistent with either economic or social strategic objectives." (p. 53)
Morris, Schindehutt, and Allen	2005	"A business model is a concise representation of how an interrelated set of decision variables in the areas of venture strategy, architecture, and economics are addressed to create sustainable competitive advantage in defined markets." (p.727)

Morris, Schindehutt, and Allen	2005	"The business model has been referred to as architecture, design, pattern, plan, method, assumption, and statement." (p.726)
Osterwalder and Pigneur	2002	"a description of the value a company offers to one or several segments of customers and the architecture of the firm and its network of partners for creating, marketing and delivering this value and relationship capital, in order to generate profitable and sustainable revenue streams." (p. 77)
Chesbrough and Rosenbloom	2002	"In the most basic sense, a business model is the method of doing business by which a company can sustain itself—that is, generate revenue. The business model spells out how a company makes money by specifying where it is positioned in the value chain." (p. 533) "The heuristic logic that connects technical potential with the realization of economic value." (p. 529)
Amit and Zott	2001	"A business model depicts the content, structure, and governance of transactions designed so as to create value through the exploitation of business opportunities." (p. 511)
Afuah and Tucci	2001	A business model can be viewed as a "system that is made up of components, linkages between components, and dynamics." (p. 4) "The method by which a firm builds and uses its resources to offer its customers better value and to make money in doing so." (p. 3)
Stewart and Zhao	2000	"A statement of how a firm will make money and sustain its profit stream over time."
Tapscott	2000	"The totality of how a company selects its customers, defines and differentiates its offerings, defines the tasks it will perform itself and those it will outsource, configures its resources, goes to market, creates utility for customers and captures profits." (p. 10)
Mayo and Brown	1999	"The design of key interdependent systems that create and sustain a competitive business."
Timmers	1998	"An architecture of the product, service and information flows, including a description of the various business actors and their roles; a description of the potential benefits for the various business actors; a description of the sources of revenues." (p. 2)

*Source:* Compiled by the Author

Analysis of the first stream of the literature reveals that "the business model is not a value proposition, a revenue model, or a network of relationships by itself; it is all of these elements together" (Zott et al., 2011, p. 1028). Authors also find that none of the research in this category analyses the relationship between any components of BM and other constructs. The examination of the second category shows interesting results too. These include notes on what BM is not.



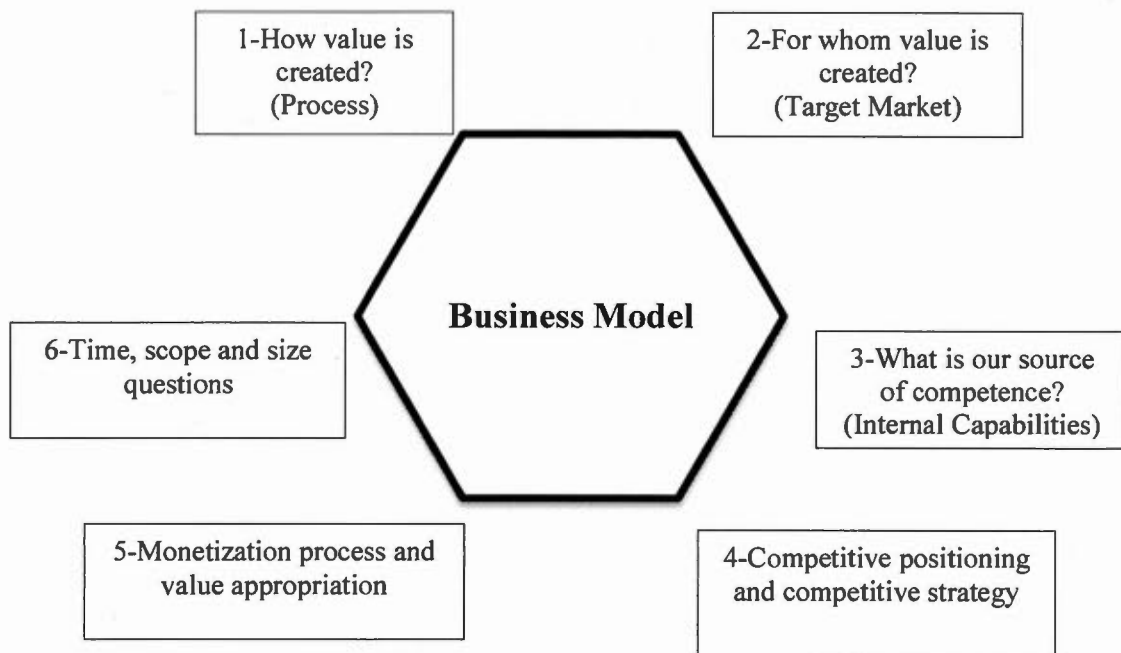
Table 2.3. The Six Main Components of a Business Model

Components	Questions	Factors
Component 1 <i>Offering-related factors</i>	How do we create value?	offering: primarily products/primarily services/heavy mix offering: standardized/some customization/high customization offering: broad line/medium breadth/narrow line offering: deep lines/medium depth/shallow lines offering: access to product/ product itself/ product bundled with other firm's product offering: internal manufacturing or service delivery/ outsourcing/ licensing/ reselling/ value added reselling offering: direct distribution/indirect distribution (if indirect: single or multichannel)
Component 2 <i>Market-related factors</i>	Who do we create value for?	type of organization: b-to-b/b-to-c/ both local/regional/national/international where customer is in value chain: upstream supplier/ downstream supplier/ government/ institutional/ wholesaler/ retailer/ service provider/ final consumer broad or general market/multiple segment/niche market transactional/relational
Component 3 <i>Internal capability-related factors</i>	What is our source of competence?	production/operating systems selling/marketing information management/mining/packaging technology/R&D/creative or innovative capability/intellectual financial transactions/arbitrage supply chain management networking/resource leveraging
Component 4 <i>Competitive strategy-related factors</i>	How do we competitively position ourselves?	image of operational excellence/ consistency/ dependability/ speed product or service quality/ selection/ features/ availability innovation leadership low cost/efficiency intimate customer relationship/experience
Component 5 <i>Economic-related factors</i>	How do we make money?	pricing and revenue sources: fixed/ mixed/ flexible operating leverage: high/medium/low volumes: high/medium/low margins: high/medium/low
Component 6 <i>Personal/ investor-related factors</i>	What are our time, scope and size ambitions?	subsistence model income model growth model speculative model

Source: Adopted from Morris, Schindehutte, and Allen (2005, p.730)

For example, “the business model does not involve a linear mechanism for value creation from suppliers to the firm to its customers” (Ibid., p.1031). This shed light on complexity associated with ‘value creation’ process as an “interconnected set of exchange relationships and activities among multiple players” (Zott et al., 2011, pp. 1031-1032). In addition, BM is not the same concept as product market strategy or corporate strategy. The concept cannot be reduced only to issues that concern the internal organization of firms; it typically span firm boundaries.

Figure 2.4. The Six Main Components of a Business Model



*Source:* Content Adopted from Morris, Schindehutte, and Allen (2005, p.730), Figure representation form is by author.

Yet, BM is a source of competitive edge. Finally, investigation of the third category of research on BM leads to insightful comments concerning the interplay between BM, innovation and technology management. Zott et al. (2011) conclude that technological innovation is important for firms, but it might not suffice to guarantee success (e.g., Doganova & Eyquem-Renault, 2009) because technology alone has no

inherent value (Chesbrough, 2007a, 2007b). Therefore, to fully realize the commercial potential of technology, a firm needs to have a unique BM which enables it to embed the technology in products and services (Zott et al., 2011).

In addition to providing a classification of the literature on BM research, Zott et al. (2011)'s analysis has led to emergence of four common themes despite the disparity of researchers' approaches to BM concept. These common themes are: 1) BM as a new unit of analysis; 2) a holistic perspective on how firms do business; 3) an emphasis on activities; and 4) an acknowledgement of the importance of value creation.

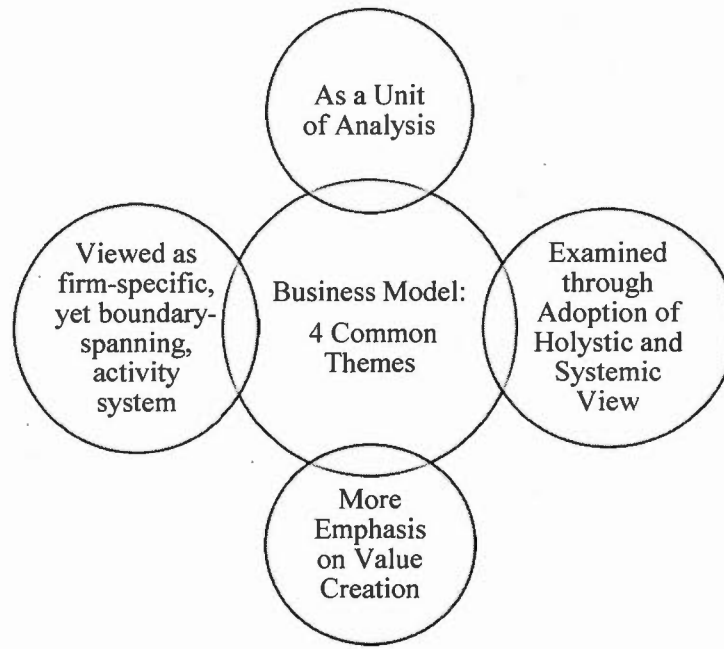
The first theme shows that BM is either implicitly or explicitly considered "as a new unit of analysis... which spans or bridges traditional units of analysis, such as the firm or the network" (Zott et al., 2011, p. 1036). The second theme highlights how scholars are shifting towards conceptualizing BM through adoption of "a holistic and systemic (as opposed to particularistic and functional) perspective, not just on what businesses do... but also on how they do it..." (Zott et al., 2011, pp. 1036-1037). Taking a holistic view, as it involves both the content as well as process of conducting business imposes the challenge of defining and operationalizing the BM construct. Moving beyond BM as representing a system-level concept, the third theme views BM as a "firm-specific, yet boundary-spanning, activity system" (Zott et al., 2011, p.1037). The literature on BM seems to support "an activity system perspective" where BM can be defined as a "system that is made up of components, linkages between components, and dynamics" (Afuah & Tucci, 2001, p. 4). Lastly, the fourth theme highlights the shift of emphasis in the literature from value appropriation or value capture to 'value creation'; while the latter is emphasized without the former being ignored (Zott et al., 2011, p.1037). This trend has even brought to the attention of those BM researchers whose primary focus has been on value appropriation by focal firm; mainly because they have realized that "the value

is created through the focal firm in concert with its exchange partners” (Zott et al., 2011, p.1037). Figure 2.5 demonstrates the four themes as discussed above.

Teece (2010) maintains that the conventional balance between customer and supplier is disturbed due to developments made in the global economy. ICTs and open global trading regimes, among others have put consumers in a more powerful position as they have more choices and various needs; “businesses therefore need to be more customer-centric” (Teece, 2010, p. 172). His perspective on BMs, indubitably, revolves with more weight around customers’ role in BM design. BM represents the “logic” and “evidence” on how a business creates and delivers value to customers. Teece’s work also outlines “the architecture of revenues, costs, and profits associated with the business enterprise delivering that value” (Teece, 2010, p. 173). Put simply, a good BM is one that can “provide considerable value to the customer and collect (for the developer or implementor of the business model) a viable portion of this in revenues” (Teece, 2010, p. 179).

Teece (2010, p. 174) identify the core elements of a BM design as follows: 1) Selecting the technology to be embedded in the product or service; 2) Determining the benefits that customers can enjoy from consuming the product or service; 3) Identifying market segments; 4) Confirming available revenue streams; 5) Designing mechanisms to capture value. Furthermore, although BM is more generic than business strategy, as put by Teece (2010), for a BM to be competitively sustainable it must survive the filters which strategy analysis imposes. This will help the BM to become hard-to-imitate.

Figure 2.5. The Four Major Themes Concerning BM Studies



Source: Contents are based on Zott et al. (2011). They are represented schematically by author.

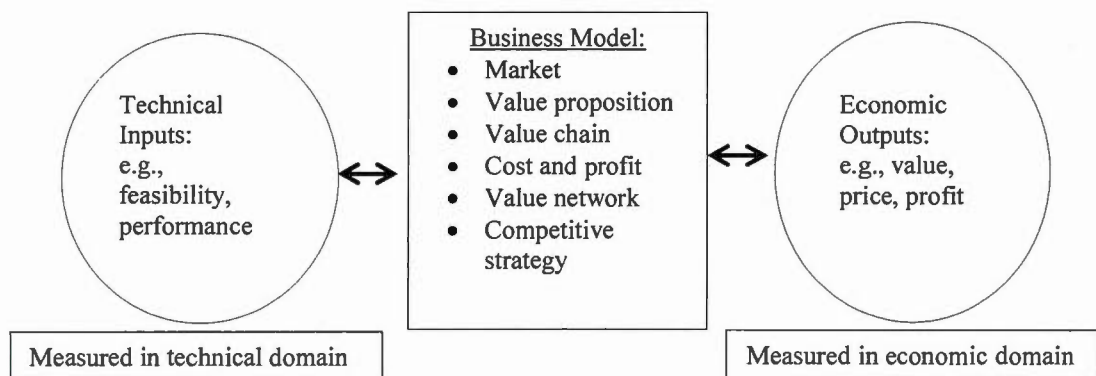
Lastly, Chesbrough and Rosenbloom (2002) explore the role of BM in capturing value from early stage technology. They view an effective BM as one that “creates a heuristic logic that connects technical potential with realization of economic value” (Ibid., p. 529). Therefore, the objective of a BM is to unlock the “latent value” from a piece of technology. Despite lack of BM conceptual clarity in the literature, the authors attempt to operationalize BM as a construct by delineating its six functions (Chesbrough & Rosenbloom, 2002, pp. 533-534):

- “articulate the *value proposition*, i.e. the value created for users by the offering based on the technology;

- identify a *market segment*, i.e. the users to whom the technology is useful and for what purpose, and specify the revenue generation mechanism(s) for the firm;
- define the structure of the *value chain* within the firm required to create and distribute the offering, and determine the complementary assets needed to support the firm's position in this chain;
- estimate the *cost structure* and *profit potential* of producing the offering, given the value proposition and value chain structure chosen;
- describe the position of the firm within the *value network* linking suppliers and customers, including identification of potential complementors and competitors;
- formulate the *competitive strategy* by which the innovating firm will gain and hold advantage over rivals”.

Figure 2.6 shows how the BM connects a technological possibility with an economic reality.

Figure 2.6. The BM Mediates between the Technical and Economic Domains



Source: Chesbrough and Rosenbloom (2002, p. 536)

### 2.5.5 Conventional BM concept poorly suited to explain distributed value creation

The review of the BM's literature makes the case for claiming that it is not a well-equipped concept to deal with the *openness*, *open strategy*, and more importantly, *open source software phenomenon*. BM, as it seems, is still very corporate-centric, focused on firm-level sustainable competitive advantage, customer segment and revenue focused, and the internal structure of the firm. Although the bulk of the literature tends to position the concept in the thinking of conventional strategic management, i.e., Porter's five forces (1980a), recently scholars have acknowledged that BM expands firm boundaries and that it is becoming more focused on value creation processes. Still, BM does not seem to have become attuned to OI or UI concepts where value creation and capture processes are not readily situated within organizational boundaries. It certainly is poorly suited to explain the core and nuances of OSSTCs.

### 2.5.6 Does BM literature embrace OI?

Why is the discussion of BM so central to researching OI field? Innovation means ideation accompanied by commercialization; according to Chesbrough (2010, p. 354), "companies commercialize new ideas and technologies through their business models". Essentially, a BM functions in two pivotal ways: it creates value and it captures (a portion of) that value (Zott et al., 2011; Chesbrough 2007a; Chesbrough, 2007b; Brandenburger & Stuart, 1996). In fact, not only BM is an indispensable part of innovation agenda of a company, but also it plays a critical role in making the whole innovation process successful or disastrous so much so that technology by itself has no single objective value. The economic value of a technology, according to Chesbrough (2010), remains latent until it is commercialized in some way via a BM. As a result, "a mediocre technology pursued within a great business model may be

more valuable that [sic] a great technology exploited via a mediocre business model” Chesbrough (2010, p. 354). For this reason we need to understand the nature of Open Business Model (OBM) in relation to OI field. In fact, “*open business models enable an organization to be more effective in creating as well as capturing value*” (Chesbrough, 2007b, p. 22). OBMs help create value by extending a firm’s arm to capture external sources of ideas and they allow greater value capture “by utilizing a firm’s key asset, resource or position not only in that organization’s own operations but also in other companies’ businesses” (Chesbrough, 2007b, p. 22).

Although several scholars have advanced our understanding of OI field of research by their scholarly attempts, there still remains rather unclear as how one makes a distinction between the concept of OI and OBM. To some, drawing a clear-cut distinction is quite confusing, therefore they suggest viewing both (OI and OBM) as one (e.g., Holm et al., 2013; Wang & Zhou, 2012; Davey, Brennan, Meenan, & McAdam, 2010, 2011; Chanal & Caron-Fasan, 2010; Gassmann, et al., 2010; Chesbrough, 2006a). For example, based on Chesbrough (2006a), the OIP requires adoption of OBMs which in turn enable firms to be more effective in creating as well as capturing value (Chesbrough, 2006a, 2007b).

Others (Alexy & George, 2013; Wang et al., 2009; Chesbrough & Schwartz, 2007), on the other hand, have made distinctions (however marginal) in the sense that OBM is *a BM based on OI principles* or that it entails adjusting a firm’s BM to OIP (Weiblen et al., 2013). In what follows, I wish to draw on the relevant literature and provide different viewpoints on the relationship between OI concept and OBM.

Among the most recent and very few focused studies conducted on OBMs, I can draw on Weiblen et al. (2013)’ study that identifies two views on OBM. One is the *open innovation view* of OBM and the other is *business model view* of OBM.



The former follows the doctrine put forth by Chesbrough (2006a, p. 107) which hinges on the debate that “companies must develop open business models if they are to make the most of the opportunities offered by open innovation”. This is further clarified by the concept of ‘opening up’ the BM “by actively searching for and exploiting outside ideas and by allowing unused internal technologies to flow to the outside, where other firms can unlock their latent economic potential” (Chesbrough, 2007b, p. 22). Following this perspective, according to Weiblen et al. (2013, p. 7), “an open business model is always accompanied by the open innovation paradigm successfully implemented in a firm’s R&D”. This is also called the R&D-centric perception of the OBM where it is built around openness in the R&D activities of a firm (Weiblen et al., 2013). The latter view of OBM, with which I agree more, takes a more liberal standpoint in that it sees all activities of a firm “as potential candidates for collaboration and thus openness” (Weiblen et al., 2013, p. 11). This means that firms can apply openness to all value creation and value capturing activities in their value chains. Therefore, as these views suggest, the concepts of OI, BM and OBM are not the same though they have overlapping constituents.

The two views have four elements in common which brings us closer to operationalizing the concept of OBM and therefore being able to draw a rather clear-cut distinction between OI and OBM. These four elements are: 1) concept of ecosystem; 2) existence of value or partner networks; 3) platforms; and 4) alliances (Weiblen et al., 2013).

In fact, OBM explicitly considers the ecosystem as a new source of value creation and capture by developing symbiotic relationships (Romero & Molina, 2011) and emphasizing inter-organizational activities (Chu & Chen, 2011). Further, creating a value network is seen as an inseparable part of an OBM (Davey et al., 2011). In addition, a firm’s technology assets form a platform which can be opened up to smaller partners by the platform owner to enable them to create additional value and

therefore influencing the industry (see Sandulli & Chesbrough, 2009b; Purdy et al., 2012; Luo & Chang, 2011). Finally, alliances— either perceived as inter-organizational legal partnership like joint ventures (Wang & Zhou, 2012) or received as the logic of managing partnerships (Enkel et al., 2009)— are an indispensable part of OBM. In short:

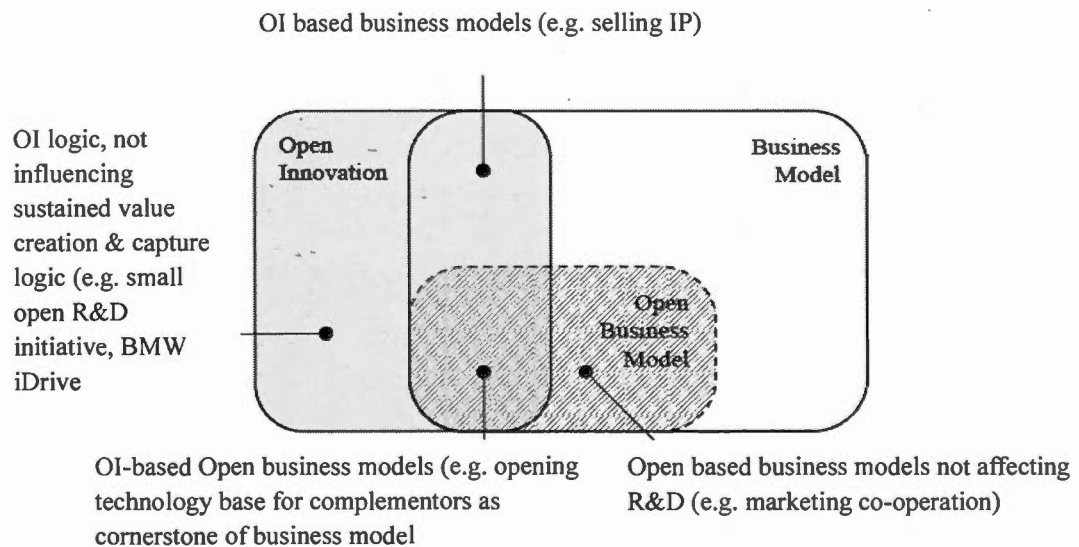
“An open business model is seen as an ecosystem-aware way of value creation and capturing. Focal firms collaborate with the ecosystem by building up value- or partner networks, platforms, or alliances and innovate their business model to make use of the emerging opportunities.” (Weiblen et al., 2013, p. 16)

Based on Weiblen et al. (2013)’s four assertions, we can better understand the differences between OI and OBM. First, an OBM includes external resources in at least one of its value creation and capturing activities. Second, in an OBM, openness in terms of collaboration should be so central to the firm’s logic of value creation and capturing that it could not be explained without it. Third, OI only constitutes an OBM if it contributes to the firm’s sustained value creation and capturing. Fourth, OI only constitutes an OBM if it leads to collaboration in the firm’s value creation and capturing activities. Figure 2.7 depicts the relationships discussed above.

Although Figure 2.7 represents the most recent and perhaps interesting analysis of the relationships between OI and OBM, one cannot fully rely on this model as it is not yet established in form of a journal article. Further, as cited earlier, many authors approximate very closely the notion of OI with OBMs. The study conducted by Weiblen et al. (2013) is a pioneering attempt to bring some conceptual clarity to the interwoven concepts of OI-based BM and OBM, and this is an ongoing line of research in the years ahead of us. According to Dahlander and Gann (2010), not any OI move constitutes an OBM; furthermore, value capturing is a major problem in this class of OIs (Dahlander & Gann, 2010). This is evident from the example put forward

by Alexy and George (2013) which regards Netscape' releasing the source code of its Navigator web browser. This has been viewed as a single-shot strategic move vis-à-vis Microsoft' dominance and it hardly forms a basis for a sustainable new BM.

Figure 2.7. Conceptual Framework of Separation and Overlap between OI, BM, and OBM Concepts



Source: Weiblen et al. (2013, p. 24)

Thus, we may accept OBM as a standalone phenomenon having certain degree of overlap with OIP. If so, it is not possible to discuss one without including the other. This connection leads to some questions such as: what are the consequences of opening the BM and what are the challenges associated with having an OBM as part of the firm's strategy.

As these issues have been a point of concern over the past decade, different scholars have touched upon different challenges as a sub-category to OBM discussion. For example, Kakaletris et al (2004) mention technical challenges while Smith et al. (2010) highlight leadership ones. Other researchers view the issue of incentivisation

(Chanal & Caron-Fasan, 2010) while some spotlight the matter of absorptive capacity (Sandulli & Chesbrough, 2009b). Laursen and Salter (2006a) caution that although the lack of openness to the external environment may reflect an organizational myopia one should consider that the degree of openness needs to be evaluated against the costs of it. Among other challenges is, how a firm should deal with the increased costs due to coordination and integration and the risk of knowledge leaks? (see West & Bogers, 2014; Dahlander & Gann, 2010; Faems et al., 2010). Still, although some scholars have directed their attention to challenges involved with OBM, Dahlander and Gann (2010) maintain that most of the existing research (in the field of OI and OBM) tends to focus on the potential benefits of openness rather than addressing the disadvantages. This has led to a ‘pro-bias’ in the literature.

One example further depicts how OBM is studied and linked to OI. Holm et al. (2013, p. 324) reiterate how literature on OI has portrayed OBM as a “contemporary and extremely useful tool” to create and capture value in collaboration with external partners. The authors conduct an empirical study on the impact of OBM in the traditional newspaper industry. This study captures the interrelations between technological discontinuity and BM adjustment in line with OIP. In fact, when the traditional BM of a matured newspaper industry is under attack mainly due to the internet as a content delivery channel (a form of technological discontinuity), the industry players open their BMs (i.e., designing OBM) to create value by including external ideas and capture value by utilizing complementary key assets of partner organizations. Therefore, disassembling the traditional BM (focused on a single delivery channel) and opening up (acquisition of content from various suppliers, crowdsourcing, outsourcing IT development, etc.) impose the challenge of critical dependency on third-party assets. In such OBM, the notion of inward-outward openness which captures: how the firm acquires third party strategic assets to create value; and how the firm shares its assets with partners to capture value, is based on OI concept (Chesbrough, 2006b). However, the scope of openness (broad vs. deep) is

one which is strongly related to the concept of boundary-spanning activities (Leifer & Delbecq, 1978). This study shows how in practice, in such an R&D unrelated industry, OBM can draw on the principles of OI while not becoming completely submerged by it. In conclusion, the authors explicitly recommend that “a more nuanced view and balanced understanding of the term openness as regards business model is needed” (Holm et al., 2013, p. 324).

To summarize our discussion on OBM, we arrive at the following concluding remarks. First, for BM to fit into OIP, it must adopt an open strategy and step outside the organizational boundaries. Put simply, BM needs to transform into OBM in order to become relevant to discussions on open and distributed innovation (in this case OSS). The discussions on open BM, therefore, are more useful in the context of OSSTCs as they have begun to tap into ecosystem way of value creation, partner networks, platforms, and alliances. All these concepts, however, are subject to detailed investigation in order to reflect the particularities of OSS technologies and their development processes. For instance, we need to understand whether or under what conditions OBM adopted by OSS firms can lead to positive network externalities and continuous value creation.

#### 2.5.7 Does OSS business models inform us about OBM concept?

Firms can more effectively create and capture value if they adopt OBMs (Chesbrough, 2007b). OBMs are justified on several grounds: inefficiencies of innovation markets, rising costs of R&D, shortening product lifecycles, and their potential to enable firms to conduct open experiments (see, e.g., Chesbrough, 2007b).

For example, somewhere between 75% and 95% of patented technologies are on the shelf left unused (Chesbrough, 2007b, p.23). This shows how markets are inefficient

in automatically assessing the true potential value of inventions, not to mention the high costs associated with transactions in the innovation markets. Furthermore, the costs of technology development (i.e. R&D) have been increasing beyond the reach of many corporations. This coupled with shortening life cycles of new products have given rise to the situation where “companies are finding it increasingly difficult to justify investments in innovation” (Chesbrough, 2007b, p. 24).

In response to these difficulties faced by the firm-level innovation process, OBM offers a pragmatic solution. Firms adopting OBMs can leverage external R&D resources to save time and money in their innovation process while still increasing their revenues by outsourcing their unrealized technological innovation (Chesbrough, 2007b).

In addition, OBM offers firms the possibility to more effectively benefit from OIP. In the aforementioned discussion on BM concept (see section 2.5.4), the research reveals the complexity of the value creation process through highlighting the interconnectedness of exchange relationships and activities among a federation of players in order to fully realize the commercial potential of technologies. An important and critical part of value creation within a BM is to experiment the results (products and services) without wasting many resources, endangering customer loyalty, firm reputation and branding of the products, to name a few. Part of this challenge is captured by Chesbrough (2007b, p. 24)’s recommendation of ‘open experiments’ as it refers to a firm’s need to “develop the ability to experiment with their business models”. This further involves firms to create processes for conducting experiments and for assessing their results in order to avoid risking any damage to their consumer brand (Chesbrough, 2007b). However, how to create real value for the end users precedes the branding issue. That is the reason so much focus is placed on the role of customers in BM discussions, and OBM is no exception.

In line with the open process of innovation (Chesbrough, 2006a) as well as creating and capturing value through OBMs, many scholars have attempted to study OSS firms so as to better understand how OBM works.

OSS—classified as a *process innovation*— is an interesting case; because, it is a process that relies on unlimited access to source code as opposed to the conventional closed and property-based approach of the commercial world (Bonaccorsi & Rossi, 2003). Similarly, Dahlander and Magnusson (2005, p. 481) maintain that “OSS challenges substantial parts of the conventional wisdom regarding the role of firms, intellectual property rights, and organizational forms”. Therefore, this unconventional attribute of OSS has put the phenomenon under research spotlight. Furthermore, as user communities are an indispensable part of OSS development, the inter-relationship between the two, too, has been in the center of attention (e.g., von Hippel, 2005); yet, much less attention has been given to the *role* of the firms in the community and how they can profit from their participation (van de Vrande et al., 2010).

Proprietary software firms are distinct from OSS firms in that their value creation process heavily relies on proprietary software codes developed by the firm’s hired software developers/employees. That is why the mainstream software firm is categorized as one which employs a closed BM. On the other hand, OSS firms’ BMs have a unique feature. They include a role for a ‘*community of software developers*’. This striking notion has been earlier touched upon by several scholars: Benkler (2002); Hertel et al. (2003); O’Mahony (2003); von Krogh et al. (2003); Dahlander and Magnusson (2005), etc.

However, to fully understand open source software business models (OSSBM), it is helpful to find out why they gained momentum. Perr et al. (2010) classifies the reasons into two: 1) market pull; and 2) technology push. Market pull reflects the

demand side of the OSSBM. For example, the need for reduced software development costs (Haeffliger et al., 2008), or enhanced product margins as well as the need for technical superiority (von Krogh, 2003). Technology push, on the other hand, captures notions such as customer acquisition, disrupting markets, and achieving socio-economic development goals (Perr et al., 2010).

Having briefly touched upon ‘why’ OSSBM has gained popularity, we need to know ‘how’ such model works. In this regard, Perr et al. (2010) address the BM paradox which exists between value creation and capture within OSS firms. Their research question concerns ‘*how can OSS firms capture the economic value that they create if the building blocks of their products can be used by competitors?*’

Perr et al. (2010) specify three BM factors that appear most responsible for success in value capture. These are: 1) *IP ownership and the license choice*; 2) *community management*; and 3) *the ability to craft a BM that is appropriate for the targeted markets and product categories*. Based on these three factors and the way OSS firms create value, Perr et al. (2010) further identify *seven types of BMs*. These are: 1) professional services and consulting; 2) support; 3) subscription; 4) dual license; 5) hybrid with proprietary extensions; 6) device; and 7) community source. I briefly elaborate on the three factors and touch upon the surface of the BM typology.

*License choice and IP ownership strategies* are important concepts as they shed light on the misconception of OSS vs. free software<sup>15</sup> as well as different types of OSS

---

<sup>15</sup>Free software licensed under the terms of the GPL is OSS; however, not all OSS is free software. The term ‘free’ refers not to the price of the software, but to the four freedoms to use, copy, modify, and redistribute the software (free or for a fee). Therefore, all OSS can be used for *commercial* purpose; the Open Source Definition guarantees this. You can even sell OSS. However, note that the word ‘commercial’ is not the same as ‘proprietary’. If you receive software under an Open Source license, you can always use that software for commercial purposes, but that doesn’t always mean you can place further restrictions on people who receive the software from you. In particular, so-called copyleft-style Open Source licenses require that when you distribute the software, you do so under the same license you received it under. (Based on Open Source Initiative website: <http://opensource.org/faq#commercial>)



licensing. They also have implications in understanding how OSSBM generates value (e.g. revenue streams).

Perhaps the two most common licensing types of OSS are GNU General Public License (GNU GPL or GPL)<sup>16</sup> and Lesser General Public License (LGPL)<sup>17</sup>. Therefore, the software licensed under the GPL terms and conditions will confer to the four core freedoms and includes a 'give-back' principle, i.e. if a firm redistributes the software, the source code and accompanying modifications made to it must also be released and revealed in transactions.

On the other hand, LGPL licensing allows software developers not only to incorporate open source libraries into their development process but also it allows for the final software product to be licensed under proprietary terms. Yet there is another licensing mode: i.e., Berkeley Software Development-BSD, which permits the appropriation of software modifications (Bonaccorsi & Rossi, 2003). Further, Perr et al. (2010) note that OSS firms can opt for dual licensing for commercial versions of their software and this allows them to capture value.

Probing further the role of community in relation to OSSBM, one can see the highlighted role of vendors as opposed to the widespread misconception that OSS is a phenomenon totally relying on their communities. In fact, based on findings of Perr et al. (2010, p. 439), "a wide array of community models exists and [it] increasingly

---

<sup>16</sup>The is the most widely used free software license, which guarantees end users (individuals, organizations, companies) the freedoms to use, study, share (copy), and modify the software. Software that ensures that these rights are retained is called free software. The license was originally written by Richard Stallman of the Free Software Foundation (FSF) for the GNU project. [http://en.wikipedia.org/wiki/GNU\\_General\\_Public\\_License](http://en.wikipedia.org/wiki/GNU_General_Public_License)

<sup>17</sup>The GNU Lesser General Public License or LGPL (formerly the GNU Library General Public License) is a free software license published by the Free Software Foundation (FSF). The LGPL allows developers and companies to use and integrate LGPL software into their own (even proprietary) software without being required (by the terms of a strong copyleft) to release the source code of their own software-parts. Merely the LGPL software-parts need to be modifiable by end-users (via source code availability): therefore, in the case of proprietary software, the LGPL-parts are usually used in the form of a shared library (e.g. DLL), so that there is a clear separation between the proprietary parts and open source LGPL parts. [http://en.wikipedia.org/wiki/GNU\\_Lesser\\_General\\_Public\\_License](http://en.wikipedia.org/wiki/GNU_Lesser_General_Public_License)

includes vendors who provide much of the core development (if not testing and documentation) staff for open source projects”. Therefore, it is needless to say that these vendors are in a position to exert their control over the software code, which later allows them to pursue a variety of BMs. Lastly, Perr et al. (2010) make distinction between horizontal and vertical markets and emphasize that newer vertical market offerings often require the subject matter expertise and engineering resources that only the vendor can provide.

Obviously, these three factors have a significant bearing on OSSBM value creation and capture sides. For example, in instances where there is lack of IP ownership and copy right assignment, a firm can devise an OSSBM based on ‘support’ (i.e., an OSSBM type) generating revenues derived from sale of customer support contracts. However, when a target marketplace is a vertical one and there is an option of crafting a product which has a licensed enterprise version, a firm can proliferate open source application and monetize through sale of proprietary versions or product line extensions (hybrid model).

Dahlander and Magnusson (2005) are also intrigued by the fact that the firm that contributes to and benefits from OSS communities does not solely control the knowledge produced to create the software which holds a central position in the firm’s product development (value creation). Therefore, the authors are interested in finding more about how OSS firm-community relationships are managed and what managerial challenges OSS firms encounter.

Employing case-study approach, Dahlander and Magnusson (2005) propose a typology of three different approaches used by firms to manage their relationships. These are: 1) *symbiotic*; 2) *commensalistic*; and 3) *parasitic*. These three approaches are not clear-cut categories but rather “steps on a continuum regarding the benefits for the communities deliberately searched by the OSS firms” (Dahlander & Magnusson,

2005, p. 487). The authors further describe the three approaches as follows. The *parasitic approach* entails the situation where a firm only focuses on its own benefits, without taking into account that its actions might harm the community. The *symbiotic approach*, on the other hand, refers to a situation where a firm co-develops hand in hand with the community. The last approach which is an intermediate way to inter-relate to the community is to benefit from the co-existence with another entity while leaving it without harm.

In addition to the three approaches, Dahlander and Magnusson (2005) highlight seven managerial challenges such as: 1) respecting the norms and values of the OSS communities; 2) using licenses in a fruitful manner; 3) attracting developers and users; 4) handling the resource consumption related to community development; 5) aligning different interests about the nature of work; 6) resolving ambiguity about control and ownership; and 7) getting acceptance for using the community-developed software in commercial applications and avoiding direct conflicts.

Based on the feedback received from interviewed firms, Dahlander and Magnusson (2005, p. 491) aptly conclude that the symbiotic approach that entails the acceptance of dual roles provides a better chance for the firm to influence the communities; yet the central issue becomes that of “how to balance a distributed knowledge system incorporating both the firm and its community, also acknowledging that the modes of control available differ widely within this system”. This returns us back to the value capture side of OBM. In this regard, the authors briefly touch upon using “*subtle means of control*” to steer the value capture. These controlling mechanisms include: 1) devoting personnel to work in or with communities; 2) creating and maintaining reputation; 3) fringe benefits; 4) the use of ‘interaction tools’; and 5) ‘selling’ development tasks. Yet, these subtle means of control practiced by four firms remain to be tested and tried by larger number of firms over time to become established as industry standards so as to be later widely adopted by firms interested in adopting

OBTMs. Also, the relative importance of these mechanisms remains a mystery to those policy makers who need convincing numbers to make a shift in policy directions. This is especially the case as the debate over patenting algorithms and business methods related to software (mainly in European context) has remained an unsettled one (see also Levin et al., 1987).

As OS communities gain more momentum with respect to OSSBTMs, in another study Dahlander and Magnusson (2008) try to understand how firms make use of these communities and how such use is associated with their business models. Authors conduct a qualitative study based on four in-depth interviews with OSS firms and inductively derive three themes: *accessing*, *aligning*, and *assimilating*. *Accessing* involves firms approach to get access to the community resources to expand their resource-base. *Aligning* reflects how a firm aligns its own strategies with those of community. *Assimilating* regards how firms assimilate with communities to integrate and share results. Each of these themes reflects an overarching strategy which involves minor tactics for implementation. For example, accessing can be done through a tactic called *establishing new communities*, which aims at attracting outsiders to work in the firm's area. Aligning requires the firm to opt for *adopting licensing practices to clarify ownership* of product developed collaboratively by the firm and its community. Finally, assimilating draws on a tactic called *devoting resources to evaluate and select source code from communities*.

Although Dahlander and Magnusson (2008) provide a rather thorough understanding on how OSS firms use OSSBM and manage their affairs to make use of communities, their work is very practice oriented and lack sound theoretical perspective underlying those themes. For example, each theme and its associate tactics draws on different or a mix of firm-specific capabilities (e.g. absorptive capability with regards to assimilating theme) which remain foggy and unattended in the course of their study.

Perhaps among other interesting academic insights which shed light on user communities<sup>18</sup>, I can note the work conducted by Dahlander and Frederiksen (2012) which highlights the relational view of innovation in user communities (here, user community for computer controlled musical instruments). In fact, there are certain truths about user communities which make them interesting in the discussion of value creation and value capture with regards to OSSBM.

For example, the vast majority of contributions within user communities are made by a few (Lakhani & von Hippel 2003) where communities portray a core-periphery structure with a cohesive subgroup of core actors and a set of peripheral actors who are loosely connected to the core (e.g. Borgatti & Everett 1999). Therefore, a member's position can change from being of pivotal to not so important in terms of its influence within a community. This has led scholars (Dahlander & Frederiksen, 2012) to view core-periphery structures in terms of a continuum and claim that a member's relative position within a core-periphery structure can have a significant bearing on innovations.

Extending this line of reasoning to OSSBM and OSS firms would suggest that the kind of position an OSS firm holds within certain communities (often more than one) can impact the direction of innovations within those communities and align them with the firm's innovation strategy. Therefore, as long as value creation and capture are involved, the discussion of OBM encompasses user communities as well.

In conclusion, although the study of the OSSBMs brings us closer to an understanding of OBM, we still do not know enough about neither OBM nor OSSBM concepts. This is more particularly so as OSS is a very specific case of OBM. OSS industry is a complex one. It comprises OS technologies, community of developers,

---

<sup>18</sup>OSS communities are in fact user communities to a large extent, however, if absolutely viewing them as such signifies an extreme view.

OS foundations, OS service providers, and all organizations that integrate OSS tools (e.g., FFmpeg, GStreamer, Linux kernel) and libraries into their value chain activities. For these reasons, a study of OSSTCs is a valid effort that can better inform us about value creation processes within OSS industry and expand our knowledge of OBM.

Moreover, 'OSS development process' can be conceptualized in different ways. It is a software development methodology (purely technical perspective). It is a strategic plan to obtain unique resources, to reduce cost and time of development to offer more competitive biddings, to build internal technical capabilities and boost in-house R&D, to build branding and attract customers, etc. (strategy and innovation perspective). It is an indispensable part of an OBM adopted by different organizations to create value through boundary spanning enabled by institutional openness (OBM enabler). It is also part of an OBM to capture a portion of the created value in different forms: monetary (e.g., revenues) and non-monetary (e.g., learning, enhancing internal R&D capacity) through formal or informal mechanisms (BM perspective). Therefore, knowing about only OSS service providers' BMs is not enough to explain all aspects of OSSBM or OBM as a larger context. We also need to study other elements of the ecosystem in relation to value creation and value appropriation processes of OSSBMs.

Furthermore, the analysis of the literature has shown a shifting emphasis on 'value creation' (without ignoring the value capture) and highlighted the key role of 'customer-focused value creation' within strategy-related discussion of BM. Both concepts of value creation and value capture are at the core of strategic management since superior value creation and the ability to capture value in the form of profits are prerequisites for building competitive advantage (Priem, 2007; Enders et al., 2008; Enders et al., 2009). However, the concept of value creation deserves special attention particularly because the process of value creation is often confused with the process of value capture (Morgan & Finnegan, 2014, p. 3). In addition, it is argued

that both value creation and capture should be viewed as distinct processes, since the source that creates value may or may not be able to capture that value in the long-term (Lepak et al., 2007). Yet, strategy management research has paid less attention to value creation than to value appropriation/capture (Nickerson et al., 2007). As a result, the task of examining how firms create and capture value with OSS is made more difficult. The situation is further exacerbated when little attention has been paid to the role of customers within OSS-related BMs. On the contrary, more focus has been on the relationship between firms and communities or interactions within communities thereby overshadowing the role of customers in joint value creation. Thus, a study of OSS needs to focus on the types, roles, and capabilities of clients in order to provide a more complete picture of innovation process and respect the holistic nature of OBM design. For example, questions like “how can OSS clients contribute to sustainable value creation process?” or “what roles can customers play in the R&D process of OSS products to reduce costs of development, time to market, and increase effectiveness of solutions?” have remained elusive in the OI as well as OSS literatures.

## 2.6 To what extent OI is a valid theory to investigate the case of OSS

I include this section mainly because there are some doubts about the explanatory power and robustness of OIP among academics and practitioners alike. The key question to ask can be, ‘is OI a fully-fledged theory?’ The answer to this question is negative and in what follows, I explain the reasons. In fact, since OI concept has been proposed by Henry Chesbrough in 2003, it has been enjoying an upward trend in popularity. However, OI concept is *not a bed of roses*! On the contrary, it has been criticized on three major fronts: a) dearth of solid theoretical underpinning; b) research shortcomings, many uncharted research domains, and dearth of external validity; and c) being built upon fuzzy pillars so much so that it has been labelled for

being “*old wine in new bottles*” (Trott & Hartmann, 2009). Thus, we cannot depend blindly follow OI teachings while studying OSSTC and OSS phenomenon in general. To err on the safe side, we may treat OSS as an independent unit of analysis. However, we should benefit from the earlier research on the relationship between OSS and OI, and try to contribute to the interface where we can.

#### A) Lack of solid theoretical underpinning

OI is viewed as a field which lacks *solid theoretical underpinning* (e.g., Lichtenthaler, 2011; van de Vrande et al., 2009) – an issue which is still valid (Vanhaverbeke & Cloudt, 2014). For instance, OI neither complies with several criteria that typically characterize management fashions (Abrahamson, 1996; Kieser, 1997) nor it qualifies as a new theory based on essential guidelines (Sutton & Staw, 1995; Zahra & Newey, 2009; as cited in Lichtenthaler, 2011, p. 79). Thus, OI is yet to be considered “a coherent new theory” (Lichtenthaler, 2011, p. 79). OI is mainly perceived as a *framework* characterizing some approaches to innovation management (Lichtenthaler, 2011, p. 80). Others (Vanhaverbeke & Cloudt, 2014) have also recognized this matter, and therefore emphasized the need for integrating management and economics theories into OI discussions in order to forward the field into a more theoretically mature state with the aspiration of transforming OI framework into a fully-fledged theory of [contemporary] innovation.

#### B) The extant research gaps and lack of external validity

OI research offers several white research spaces (i.e., “uncharted research domains”) so much so that it still needs to be studied in *a wide variety of settings* (van de Vrande et al., 2010, p. 229). West and Gallagher (2006) emphasize the need to extend OI research to *different levels of analysis*. OECD (2008) highlights the role of OI-related



*public policy implications*. Still others underline the necessity to direct OI's focus on *SMEs* and SMEs in *low-tech industries* (van de Vrande et al., 2010, 2009). This is mainly because most research on OI has been focused on high-tech multinationals (van de Vrande et al., 2009; Chesbrough, 2003a); while research has shown us that "lessons learned from open innovation in large firms are not readily transferrable to the context of SMEs..." (van de Vrande et al., 2010, p. 227).

Moreover, OI suffers from several other research paucities. For example, we still need to obtain a clearer understanding of "OI's characteristics" as well as "practices and tools" for managing its processes (Lichtenthaler, 2011). Others (Lichtenthaler, 2011; van de Vrande et al., 2009; Gassmann & Enkel, 2004) emphasize the dominance of (or exaggerated role of) *qualitative research approaches* compared to quantitative methodologies in exploring the phenomenon. This methodological imbalance has several repercussions with the most important being the issue of *external validity* of OI-focused studies. Thus far, only few studies have explored openness using large-scale datasets (e.g., Laursen & Salter, 2006a). Therefore, exploring other empirical settings is important for achieving external validity (Chesbrough & Crowther, 2006); i.e. increasing generalizability of the results. This will also satisfy policy makers' need for statistics to make sure about OI's relevance to larger business populations.

### C) OI's fuzzy pillars: Reviewing the past criticism on OI

OI concept – which has been celebrated for a little more than a decade by its proponents within academia (e.g., Gassmann, 2006; Henkel, 2006; West & Gallagher, 2006; Dittrich & Duysters, 2007; van de Vrande et al., 2009; Enkel et al., 2009;

Gassmann et al., 2010) and industry (e.g., Philips; Procter & Gamble Co; Dodgson et al., 2006) – now seems to enjoy an upward trend in popularity<sup>19</sup>.

Earlier in this chapter, I have attempted to review the OI concept mainly based on the major research works conducted on and around the topic. Mostly, I have cited scientific research efforts that have attempted to further clarify OI concept and solidify its theoretical pillars based on the implicit assumption that: it is a novel conceptualization of modern innovation paradigm; in other words, the way forward in the years to come. However, not all academics in the field of R&D and innovation management, or even practitioners for that matter, share the same positive and encouraging sentiment towards OI. On the contrary, despite dearth of criticism officially published in academic journals, there are three academic papers that have expressed criticism (albeit to different degrees) over the concept; and they are noteworthy to mention. These are: a) Trott and Hartmann (2009)'s; b) Mowery (2009)'s; and c) Groen and Linton (2010)'s study.

First, Trott and Hartmann (2009, p. 715) claim that Henry Chesbrough has created “a false dichotomy by arguing that open innovation is the only alternative to a closed innovation model”. Thus, the authors believe that the works of Chesbrough (2003a, 2003c) in which he presents the six principles of closed innovation build the case for “a straw man argument, which misrepresents the true position of innovation management today” (Trott & Hartmann, 2009, p. 716). Consequently, the critical authors draw on long standing literature in the field of R&D and innovation management to show that the core concept of OI represents “*little more than the repackaging and representation of concepts and findings over the past forty years within the literature on innovation management*”(Ibid., p. 715). For example, they cite Rothwell and Zegveld (1985)'s “Reindustrialization and Technology” in order to

---

<sup>19</sup> For a review of the concept, please read Chesbrough and Bogers (2014). In their work, the authors have provided an updated review of the concept of OI and its current popularity.

demonstrate that well over twenty years ago, others have emphasized “the need for external linkages within innovation process” (Trott & Hartmann, 2009, p. 716). In short, Trott and Hartmann (2009) believe that Chesbrough has not given the past literature its due while introducing the concept of OI.

Second, Mowery (2009)’s paper also provides traces of criticism concerning OI concept, albeit to a more limited extent. His paper reviews the development of industrial R&D in the United States during the postwar period (i.e., since 1945). The author believes that the structure of industrial R&D has been changing considerably since 1985. Mowery (2009, p. 1) claims that the change has not been towards “creating an entirely novel system” yet “this restructuring has revived important elements of the industrial research ‘system’ of the United States in the late 19<sup>th</sup> and early 20<sup>th</sup> centuries”. His analysis aptly shows that “many of the elements of the ‘Open Innovation’ approach to R&D management (Chesbrough 2003a, 2006b) are visible in this earlier period” (p. 1).

“Indeed, the 1945–1985 period in the historical development of industrial R&D in the United States, which was characterized by large central corporate research facilities that sought to span the continuum from fundamental research to development, ultimately may prove to be a departure from a structure that for much of its existence included both inter-institutional linkages and a market for intellectual property to support industrial innovation”.

Thus, early 1970s marks the beginning of the structural change in the pattern of US industrial R&D where “large corporations reduced or eliminated their central R&D laboratories, increasing their *reliance on external sources of R&D and knowledge*, such as *universities, interfirm alliances, licencing transactions*, and acquisitions of other firms” (Mowery, 2009, p. 13; Emphasis added).

Further, Mowery (2009) places an emphasis on the key role of ‘public policy’ (more particularly, the antitrust policy) as a major influence on the evolution of the US industrial R&D system during the past 125 years. He emphasizes the role of public R&D spending as another aspect of major influence of public policy in the post-1945 period.

In short, Mowery (2009)’s research shows the emergence of what later has been termed “open innovation” in the US industrial R&D by highlighting 1985 as the turning point in the US industrial R&D pattern and influence of public policy.

Third, Groen and Linton (2010) criticize the OI concept from a *terminology perspective*. They claim that OI corresponds to the definition of supply chain management<sup>20</sup> resting upon the core argument that “supply chain management focuses on the creation of value by reaching beyond the traditional borders of a firm including suppliers, customers and other stakeholders” (Groen & Linton, 2010, p. 554). Therefore, arguing so, using OI in the realm of technology innovation management and supply chain management in other fields, the authors pose the question: “are we creating false barriers that inhibit communication between different groups of academic?” (Ibid., p. 554).

The above mentioned three critics have been very briefly attended to within the latest work of Chesbrough and Bogers (2014, pp. 22-24). To respond to Trott and Hartmann’s (2009) paper, Chesbrough and Bogers (2014) build their argument over the “erosion factors”<sup>21</sup> introduced by Chesbrough (2003a). In their defense, Chesbrough and Bogers (2014, p. 22) claim that “Chesbrough (2003a) identified

---

<sup>20</sup> Supply Chain Management defined as “key business processes from end user through original suppliers that provides products, services, and information that add value for customers and other stakeholders” (Lambert & Cooper, 2000).

<sup>21</sup> Erosion factors: increased mobility of workers, more capable universities, declining U.S. hegemony, and growing access of startup firms to venture capital (re-stated in Chesbrough & Bogers, 2014, p. 22; Chesbrough, 2003a)

erosion factors that influence the conditions under which innovation takes place have changed, giving rise to a new paradigm in which firms need to be and benefit more from being open for innovation. This does not imply that individual elements of open innovation were absent in the earlier paradigm, but rather that they now combine to form a new paradigm to manage innovation". Furthermore, the authors mention that in their version of the book (2003a), Chesbrough did cite prior literature and gave credit to pioneering authors.

Further, Chesbrough and Bogers (2014, p. 23) perceive Mowery's (2009) work as "regrettably" incomplete, "with little notice taken of any of the evidence or the analysis offered in the Chesbrough (2003a)". Chesbrough and Bogers (2014, p. 23) claim that Mowery's (2009)'s research is in sharp contrast with Trott and Hartmann's (2009), in that the former emphasizes change in the "industrial innovation" while the latter claims "little has changed". Then, again, the authors turn the spotlight on "erosion factors" and claim that the work of Mowery provides yet "further evidence for some of the erosion factors noted by Chesbrough (2003a)." Thereby, turning the criticism into more credit for OI, authors claim that the erosion factors accentuate the need for OI. Chesbrough and Bogers (2014, pp. 23-24) state that:

"Mowery's critique also overstates his argument. There is indeed some continuity between the innovation systems of a century ago and the systems of today. The large-scale industrial research laboratories that arose during the last century have receded. But to state that we have returned to innovation conditions of a century ago overlooks a great deal that is new. The roles of startup firms, of venture capital, the growth of federally funded university research...all of these differ substantially from the period of the second industrial revolution..."

Furthermore, Chesbrough and Bogers (2014) claim that their work is based on primary qualitative data while Mowery's (2009) research is based on secondary data analysis.

Finally, Groen and Linton' (2010) critique has been attended by Chesbrough and Bogers's (2014) as well as Badawy's (2011) responses. Chesbrough and Bogers (2014, p. 24) claim that OI is different from supply chain management because first it specifically deals with "the creation of new products, services and processes" ; and second, OI "covers a much broader range of possible stakeholders, which provides value in using the term open innovation as distinct from supply chain management". Similarly, Badawy (2011, p. 66) perceives OI as "a business model and a paradigm" while he defines supply chain management as "essentially a tool or component of a system for achieving efficiency in the supply chain". Therefore, to Badawy, OI and supply chain management are two "remarkably different" issues concerning their '*focus*' and '*rationale*'. As far as issue of '*focus*' is considered, Badawy (2011, p. 66) notes that:

"The focus in open innovation is on searching for new ideas, products, or services as a basis for collaboration and partnerships among organizations, companies, entrepreneurs, and other. However, the focus in supply chain management is primarily on cost-cutting and efficiency through establishing a network of interconnected system spanning all movement and storage of raw materials, work-process inventory, and finished goods from point of origin to point of consumption (supply chain)."

The rationale behind the two concepts is also different. As far as issue of '*rationale*' is considered Badawy (2011, p. 66) notes that:

"The rationale behind the supply chain management as a strategic set of tools relates essentially to the coordination of the traditional business functions. On the other hand, the open innovation paradigm embraces the non-traditional

activities relating to creativity, invention, innovativeness, and product/ service differentiation from the currently existing and accepted mold. As such, “newness” is the name of the game.”

In view of the above, it seems obvious that although OI has received very few published critiques, OI founders have not been successful in regards to providing plausible responses *as why OI should be considered a new paradigm for R&D and innovation management*. Nor have they embedded their counter-arguments in theoretically strong molds (see Chesbrough & Bogers, 2014). Their major focus point in their counter-arguments mainly revolves around the *erosion factors* that are believed to have given rise to the emergence of the concept. Thus, we need to review their relevance to be able to appreciate their explanatory power.

#### D) Reviewing Chesbrough’s “erosion factors”

There are two major tenets underpinning OI framework: 1) Six contrasting principles of closed and open innovation (Chesbrough, 2003a, 2003c); and 2) Erosion factors (Chesbrough, 2003a, 2003c). Previously, Trott and Hartmann (2009), relying on a literature review, showed that in principle closed innovation never exist. They criticized OI by building arguments that could refute the six principles of OI paradigm. However, Chesbrough and Bogers (2014) focused on erosion factors and their significant influence on shifting the closed innovation paradigm towards an open one in order to respond to major parts of Trott and Hartmann’s (2009) arguments. So, what are these factors in Chesbrough’s language?

“Toward the end of the 20<sup>th</sup> century, though, a number of factors combined to erode the underpinnings of closed innovation in the United States. Perhaps chief among these factors was the dramatic rise in the number and mobility of knowledge workers, making it increasingly difficult for companies to control their proprietary ideas and expertise. Another important factor was the

growing availability of private venture capital, which has helped to finance new firms and their efforts to commercialize ideas that have spilled outside the silos of corporate research labs. Such factors have wreaked havoc with the virtuous cycle that sustained closed innovation...” (Chesbrough, 2003c; p. 36).

First, Chesbrough’s (2003c) argument concerning erosion factors seems to be limited to the geography of the United States at best, as stated by himself. In addition, based on classification made by ‘2015 Index of Economic Freedom’<sup>22</sup> which divides 186 countries into six categories (free, mostly free, moderately free, mostly unfree, repressed, and not ranked), only five countries are listed under free market economy category; and only 30 are classified as mostly free. The USA and Canada, as representative of North America, are two countries listed as mostly free concerning their economies.

Furthermore, countries which rank high on ‘Knowledge Economy Index’ (four pillars: Economic Incentive and Institutional Regime-EIR, Education, Innovation, and Information and Communications Technologies-ICT) form only a limited number of countries and not all are homogenous on each of the dimensions. For example, even USA, which is considered among knowledge economies in the world, does not rank among the top 10 on education, EIR, and ICT dimensions<sup>23</sup>. Therefore, highlights of the US economy such as government sound public policies (e.g., antitrust, Bayh-Dole IP policy for university ownership, etc.), and high mobility of knowledge workers as eroding factors need to be taken with a pinch of salt within the US context, let alone to other countries.

---

<sup>22</sup> Please refer to online source: 2015 Index of Economic Freedom; available at <http://www.heritage.org/index/ranking>.

<sup>23</sup> please see the report: “knowledge economy index (kei) 2012 rankings” published on the world bank’s website available at <http://web.worldbank.org/wbsite/external/wbi/wbiprograms/kfdlp/extunikam/0,,contentmdk:20584250~pagepk:64168427~pipk:64168435~thesitepk:1414721,00.html?>



Second, Gompers and Lerner (2001, p. 164) argue that “many policymakers have a perception that venture capital organizations have had much to do with the rising leadership of U.S. firms in high-technology industries... But demonstrating a causal relationship between the presence of venture capital investment and innovation or job growth is a challenging empirical problem.” They claim that both venture capital funding and patenting could be positively related to a third unobserved variable such as “*arrival of technological opportunities*”. Others have tried to establish causality between venture capital funding and innovation; but they have not arrived at confirming evidence.

For example, Hellmann and Puri (2000), paying only modest attention to causality concern, found that venture capital may be particularly important for innovative firms. Gompers and Lerner (2001) interpret this finding in light of the possibility that more innovative firms select venture capital for financing, rather than venture capital causing firms to be more innovative. In another study, Kortum and Lerner (1999; see also 2000) tried to search for patterns on aggregate industry level as opposed to only firm level. Having cared for causality concerns, their results showed that venture funding did not have a strong positive impact on innovation. Therefore, the issue remains open concerning the role of venture capital in an economy’s overall process of innovation (Gompers & Lerner, 2001). Thus, even if a closed innovation model ever existed, the role of venture capital in transforming the landscape of innovation remains questionable. However, one cannot deny the fast paced growth in the venture capital industry in the United States (e.g., Gompers, 1996; as cited in Gompers & Lerner, 2001 ) and its importance for providing financial support to entrepreneurial firms (e.g., Gompers & Lerner, 2001).

Third, there has been an increase in the number and mobility of knowledge workers in the global village of today. However, this does not mean high-tech organizations have lost control over their strategic intangible assets. It is in fact still a debatable

issue within the field of research on social capital. Perhaps, the better way of formulating the issue is to look at intangible assets from a social capital perspective in which sense there is a debate over the locus of ownership of interorganizational relationships. Therefore, the key work here is 'relationship' and not necessarily mobility of knowledge workers in real space-geography issue. Most recently, based on the assumption that interorganizational relationships rely on their individuals, Sorenson and Rogan (2014) attempted to settle the argument, "who owns such relationship (employee or employer) and who can benefit from that". The authors found three factors increasing the odds associated with individual ownership. These are: a) the extent to which the resources valued by alters belong to the individual (rather than the organization); b) the degree to which alters feel greater indebtedness to the individual than to the organization; and c) the extent to which relationships involve emotional attachment. I mention this example to highlight the details involved with locus of ownership and that it cannot be simplified to a mere statement that because individuals can move around more often, therefore, the innovation regime has simply changed.

## 2.7 Conclusion

In this chapter, I have reviewed and analyzed the relevant literatures on OI, open strategy, BM, OBM, and OSSBM. Based on the review, we can consider 'openness' as an essential condition to adopt OIP, but it is not sufficient to remain successful and relevant to the competitive arena. In fact, firms need to change their BMs and adopt OBM to benefit from OIP. Furthermore, OI has not matured into a fully-fledged theory, mainly viewed as a framework in progress. Prior research also shows that OI, BM, and OBM concepts have been studied dominantly from a firm-centric perspective to the extent that user communities, ecosystems, and network of partners have received less attention. If we factor in the paucity of research on interactive and

reciprocal nature of coupled OI processes as well as lack of focus on 'value creation processes' within OBM context, the study of OSSTCs and OSS development process become a valid investigation that can add significant value to current understanding of these fields.

## CHAPTER III

### PERSPECTIVES ON OPEN SOURCE SOFTWARE DEVELOPMENT

#### 3.1 Introduction

The objective of this research is to better understand the nature of OSS technological collaborations (OSSTC) and OSS collaborative R&D and innovation process (RDIP) at heart of which lies co-value creation processes. Ultimately, I intend to develop a theory that explains and relates key participants' motivations, roles, and interrelationships to success and sustainability of OSSTC— while accounting for inherent particularities of OSS technology. In this Chapter, I first distinguish between 'free/libre' software and OSS and further shed light on the most recent characterization of OSS which captures its metamorphosis into a '*strategic and commercial*' software product. Then, I present a brief argument concerning the significance of OSS and its development process as an appropriate case for researching 'collaborative' technology development process. Next, I discuss the five inherent particularities of OSS technology that lead to building a special case to study OSS RDIP as an open and distributed innovation process. I, then, revisit the existing theoretical perspectives on OSS development; these include: motivations to engage in invention and development process; measures of OSS success; as well as governance, organization and innovation process of OSS. I close this chapter by presenting a conclusion.

### 3.2 The origin of OSS: The free/libre software movement and beyond

The origin of the ‘free software movement’ dates back to about 1984/1985 in which period of time Richard Stallman, a researcher in the field of computer science at the MIT’s Artificial Intelligent Laboratory in Boston, resigned from MIT and founded the Free Software Foundation (FSF) in 1985. The FSF is believed to have been shaped as a fight against the monopoly power of the software giants like Microsoft. For example, AT&T’s decision to turn the UNIX operating system into a commercial proprietary program was amongst the triggers that induced Stallman to react and embark on inventing GNU (*Gnu is Not UNIX*) operating system in 1983. About a decade later, in 1992, Linus Torvald’s Linux kernel filled the missing component of GNU and made it fully operational.<sup>24</sup>

FSF is a non-profit organization with a worldwide mission to promote computer user freedom and to defend the rights of all free software users (see, FSF). In one of the recent speeches of Stallman (TED<sup>x</sup> talk at “FREEDOM (@ digital age”) in April 2014), the man, repeatedly, explains the fundamentals of the free software movement. He perceives free software as “the first battle in the liberation of cyberspace”. As computers and computing lie at the core of human daily activities, the situation begs the question: “Who controls your computer?” “Is it you? Or, is it some big company that’s really controlling it?” (Stallman’s speech, 2014)<sup>25</sup>. Free software phenomenon is therefore grounded in social, philosophical and political debates. “Free software” is defined as:

---

<sup>24</sup> For further information and reference please see: a) FSF’s website: <https://www.fsf.org/>; b) von Hippel and von Krogh (2003); c) Bonaccorsi & Rossi, (2006); d) Richard Stallman’s Speech in April 2014 in the event: TED<sup>x</sup>Geneva2014: Freedom (@ digital age). Information is available in the video which can be retrieved from <http://www.tedxgeneva.net/tedx-geneva-2014-freedom-digital-age/>.

<sup>25</sup> Richard Stallman’s speech in April 2014 in the event: TED<sup>x</sup>Geneva2014: Freedom (@ digital age). Information is available in the video, which can be retrieved from <http://www.tedxgeneva.net/tedx-geneva-2014-freedom-digital-age/>.

“... software that respects users’ freedom and community. Roughly, it means that **the users have the freedom to run, copy, distribute, study, change and improve the software**. Thus, “free software” is a matter of liberty, not price. To understand the concept, you should think of “free” as in “free speech,” not as in “free beer”. We sometimes call it “libre software” to show we do not mean it is gratis.”<sup>26</sup>

Further, free software denotes that its users can have “four essential freedoms”; these are:

- I. The freedom to run the program as you wish, for any purpose (freedom 0).
- II. The freedom to study how the program works, and adapt it to your needs (freedom 1). Access to the source code is a precondition for this.
- III. The freedom to redistribute copies so you can help your neighbor (freedom 2).
- IV. The freedom to improve the program, and release your improvements to the public, so that the whole community benefits (freedom 3). Access to the source code is a precondition for this.<sup>27</sup>

Although the free software doctrine has its own proponents and forms the bedrock of Open Source (OS) movement, it did not evolve into mainstream immediately and industry was especially suspicious of it. As cited in von Hippel and von Krogh (2003, p. 2010), “in 1998, Bruce Perens and Eric Raymond agreed that a significant part of the problem resided in Stallman's term “free” software, which might understandably have an ominous ring to the ears of business people”. This gave birth to the OS movement to be founded by prominent hackers (Perens, 1998, as cited in von Hippel & von Krogh, 2003, p. 210). Although core licensing practices which have been upheld by the free software movement are incorporated into OSS (see, von Hippel & von Krogh, 2003), to Stallman they are not the same. According to FSF:

---

<sup>26</sup>Definition is based on GNU Operating System’s website founded by Richard Stallman: <http://www.gnu.org/>

<sup>27</sup> These four essential freedoms are based on GNU Operating System’s website founded by Richard Stallman: <http://www.gnu.org/>

“The fundamental difference between the two movements is in their values, their ways of looking at the world. For the Open Source movement, the issue of whether software should be open source is a **practical question** [emphasis added], not an ethical one. As one person put it, “Open source is a **development methodology** [emphasis added]; free software is a social movement.” For the Open Source movement, non-free software is a suboptimal solution. For the Free Software movement, non-free software is a social problem and free software is the solution.”<sup>28</sup>

The term ‘*open source*’ (OS) was coined on February 3<sup>rd</sup> 1998 in Palo Alto, California to highlight the “pragmatic” and “business-case ground” associated with the OS phenomenon. The business sense that originally motivated Netscape to release their code illustrated a valuable way to engage with potential software users and developers, and convince them to create and improve source code by participating in an engaged community. The newly coined term, gave the phenomenon an identity that liberated it from philosophical and political tints associated with its predecessor: free software<sup>29</sup>. Similarly, Fitzgerald (2006, p. 590) mentioned that the term free software could lead to “the common misperception” that contributors (individual and organizations) could not make money with free software. Today, OS refers to open source projects, products, or initiatives that embrace and celebrate *open exchange, collaborative participation, rapid prototyping, transparency, meritocracy, and community development*.<sup>30</sup> As the present research is concerned with ‘*open source software*’ (OSS), the focus will be limited to OS ‘software’ as defined by Open Source Initiative (OSI):

“Open source software is software that can be freely used, changed, and shared (in modified or unmodified form) by anyone. Open source software is

---

<sup>28</sup> Quoted from the article: Why “Free Software” is better than “Open Source” which can be retrieved online from GNU operating system’s website sponsored by Free Software Foundation at: <https://www.gnu.org/philosophy/free-software-for-freedom.html>

<sup>29</sup> See the Open Source Initiative Website: <http://opensource.org/history>

<sup>30</sup> The open source perspective is adopted from OpenSource.com: <http://opensource.com/>

made by many people, and distributed under licenses that comply with the Open Source Definition.”<sup>31</sup>

Other academic scholars (e.g., O’Mahony, 2003; Lerner & Tirole, 2002) have also relied on the OSI for providing a sound definition of OSS, mainly because the OSI, a non-for profit organization founded in 1998, is dedicated to promoting OSS and is the founding father of its definition. Furthermore, the distribution of OSS must comply with certain criteria; namely<sup>32</sup>:

1. Free redistribution:

The license shall not restrict any party from selling or giving away the software as a component of an aggregate software distribution containing programs from several different sources. The license shall not require a royalty or other fee for such sale.

2. Source code:

The program must include source code, and must allow distribution in source code as well as compiled form. Where some form of a product is not distributed with source code, there must be a well-publicized means of obtaining the source code for no more than a reasonable reproduction cost preferably, downloading via the Internet without charge. The source code must be the preferred form in which a programmer would modify the program. Deliberately obfuscated source code is not allowed. Intermediate forms such as the output of a pre-processor or translator are not allowed.

3. Derived works:

---

<sup>31</sup> The definition of OSS is based on Open Source Initiative’s Official Website: <http://opensource.org/>

<sup>32</sup> The criteria for distribution of OSS is based on Open Source Initiative’s Official Website: <http://opensource.org/>



The license must allow modifications and derived works, and must allow them to be distributed under the same terms as the license of the original software.

4. Integrity of the author's source code:

The license may restrict source-code from being distributed in modified form only if the license allows the distribution of “patch files” with the source code for the purpose of modifying the program at build time. The license must explicitly permit distribution of software built from modified source code. The license may require derived works to carry a different name or version number from the original software.

5. No discrimination against persons or groups:

The license must not discriminate against any person or group of persons.

6. No discrimination against fields of endeavor:

The license must not restrict anyone from making use of the program in a specific field of endeavor. For example, it may not restrict the program from being used in a business, or from being used for genetic research.

7. Distribution of license:

The rights attached to the program must apply to all to whom the program is redistributed without the need for execution of an additional license by those parties.

8. License must not be specific to a product:

The rights attached to the program must not depend on the program's being part of a particular software distribution. If the program is extracted from that distribution and used or distributed within the terms of the program's license,

all parties to whom the program is redistributed should have the same rights as those that are granted in conjunction with the original software distribution.

9. License must not restrict other software:

The license must not place restrictions on other software that is distributed along with the licensed software. For example, the license must not insist that all other programs distributed on the same medium must be open-source software.

10. License must be technology-neutral:

No provision of the license may be predicated on any individual technology or style of interface.

The definition of OSS and the criteria governing its creation and distribution ensure that OSS development process incorporates notions of: *openness, knowledge exchange and collaboration, transparency* and *porous innovation boundaries* into the innovation process.

Recently, OSS has become the subject of much commercial interest probably because it seems more promising in coping with “the core issues of the software crisis, namely that of software taking too long to develop, exceeding its budget, and not working very well” (Feller & Fitzgerald, 2000, p. 58). To certain extent, as OSS proponents claim, OSS methodology, as a particular way to software development, has even the capacity to compete successfully, and perhaps in many cases replace traditional commercial development methods (Mockus et al., 2002). Beyond this, the landscape of OSS development (OSSD) and innovation process has undergone a transformation which according to some scholars (Fitzgerald, 2006, p. 587) have led to OSS phenomenon metamorphosing into “a more mainstream and commercially viable form” dubbed as “OSS 2.0”. Such metamorphosis has influenced

'development process' of OSS as well as its 'product domains' through leaving its footprint on the four phases of software development process (i.e., planning, analysis, design, and implementation) (Fitzgerald, 2006). Among major changes concerning its development process and end-product are the following, as emphasized by Fitzgerald (2006).

OSSD process has become "less bazaar-like" due to more emphasis on "strategic planning" associated with the planning phase of development – which was not initially the case in early OSSD (Fitzgerald, 2006, p. 594). In fact, as Raymond (1999) notes, the planning phase of OSSD has been mainly ignited by "an itch worth scratching" where developers are mainly users of the software being developed. OSS becomes more commercially demanding and private firms consider how best they can gain an edge from OSS– competing with its proprietary counterparts (e.g., Linux operating system, Apache web server, Mozilla web browser, MySQL database management system, etc.). Now, there is more strategic planning involved in OSSD process. This trend has been followed by more complexity being associated with software "analysis" and "design" as the OSS as a product "is moving from development as back-office, invisible infrastructure to front-office, highly visible deployment of IS applications in vertical domains" (Fitzgerald, 2006, p. 591). Thus, issues like "rigorous project management" to create professional products; "branding" of OSS products to increase trustworthiness leading to achieve market leader status; offering a "whole-product concept of a market-driven business approach" to deliver a solution to customers in terms of products and services have become the distinguishing features of the new mode of OSS as OSS 2.0 (see, Fitzgerald, 2006).

Therefore, the focus of this research is on the OSS as the technology (e.g., Odoo, i.e., an open source ERP) that is designed and developed to satisfy a market demand; one that deals with commercial clients. This research is different from previous research that focuses on understanding how OSS functions (e.g., von Krogh & von Hippel,

2006; Bonaccorsi & Rossi, 2003), or, on how the commercial firms either in form of OS service providers or commercial software companies engage in OSS development process to gain an edge (West & Gallagher, 2006). The distinction is the integration of the 'clients' of commercial entities as stakeholders of this open and distributed development and innovation process; and further emphasizing their influential role in developing, and sustaining the OSS 2.0, as I call it 'strategic and commercial' OSS.

### 3.3 Significance of OSS as a case for open and collaborative technology development

OSS refers to a 'development methodology' where geographically dispersed programmers collaborate to jointly produce software using virtual collaboration tools (West & O'Mahony, 2005). It is perceived as "probably the most well-known case" when collective aspect of new software development and innovation is emphasized among "geographically dispersed individuals" (Dahlander, Frederiksen, & Rullani, 2008, p. 115). It also is an "extremely successful exemplar of globally distributed development" (Fitzgerald, 2006, p. 594).

On the one hand, the two key features of OSS technology development process such as: 1) collaborative development (using donated labor); and 2) shared rights to use the resultant technology, are among the commonalities between open source and open innovation strategy (see, e.g., West & Gallagher, 2006). OIP emphasizes collaborative or pooled R&D efforts to come up with new products and services (see, e.g., Chesbrough, 2003a, c; 2007b). Therefore, OSS has been viewed as "a great exemplar of open innovation because of the shared rights to use the resulting technology as well as the collaborative development of the technology" (West & Gallagher, 2006, p. 322) (see also Morgan & Finnegan, 2010, p. 76; Lundell & van der Linden, 2013). More specifically, as OSS is gaining more and more momentum and is becoming an indispensable part of software industry, OI has become a more

relevant framework to study how private enterprises have ventured into exploiting the opportunities offered by OSS sector (West & Gallagher, 2006).

On the other hand, as OSS development methodology enables participants to: a) find optimum solutions to complex programming issues; b) reduce cost and duration of software R&D; c) mitigate the risk associated with software development and allay uncertainty with new software inventions; and d) learn know-how and capabilities which cannot be easily codified and learned otherwise, it shares an uncanny resemblance with strategic/technological alliances/collaborations. For example, Hagedoorn's (1993) seminal paper provides a large number of studies that shed light on motives underlying strategic technology partnering. These motives or underlying rationales, to name just a few, include: reducing the technological complexity and achieving technological synergies; reducing, minimizing and sharing the uncertainty associated with R&D projects; reducing and sharing the costs of R&D; capturing of partner's tacit knowledge of technology; reducing the period between invention and market introduction (see Hagedoorn, 1993, p. 373).

Therefore, although OSS breaks away from "established assumptions about innovation" (von Krogh & von Hippel, 2006, p.975), we must be cautious and ask the questions: Is OSS really a poster child of OI concept? Or, is it a next generation of *technological alliances* within OSS industry? More specifically, as OI is not yet a theory, we need to be careful not to use a theoretical container (i.e., OI concept) which cannot encompass the contained! For this reason, I tend to position OSS technological collaborations (OSSTC) between OI (best perceived as a BM) and technological alliances as understood within the literature of strategic alliances. To embark on this journey, first we need to better understand the theoretical perspectives on OSS and highlight its blind spots.

### 3.4 The five inherent particularities of OSS technology

OSS is a special case technology because of having five inherent particularities. First, it is *modular*. Modularity refers to the fact that parts or processes of an innovation may be broken down into more essential parts—which can be worked on independently. It is largely a property of the OSS technology. In fact, globally dispersed individuals can develop and improve components, further assemble them together and later integrate them into the larger system (i.e., code base; e.g., the Linux kernel) in order to make them function together. Other innovations in contrast (e.g., biomedicine) may be more difficult to separate into smaller building blocks because they are more intricately tied to other components which are difficult to separate. Modularity thus facilitates open development strategies.

Second, OSS technologies embody an *evolutionary character*. Several factors contribute to this attribute. One is the ‘constantly evolving network of technologies’ to which a wide range of OSS technologies contributes. For instance, Microsoft Azure Cloud technology (proprietary owned technology) relies on Docker (OS project licensed under Apache License 2.0)<sup>33</sup>. Another is ‘constantly evolving and changing needs’ of end-users in the market place. As Weber (2004, p. 77) asserts, “prediction is actually the enemy in software testing”. Third, software technology demonstrates the traits of public goods (non-rivalry and non-exclusivity) where there is no case of tragedy of commons. For instance, if I download and use VLC<sup>34</sup>, a free and OS application, I will not exclude others to do so, nor will I reduce the availability of VLC to others. Having a public good’s quality further leads to augmented positive network externalities.

---

<sup>33</sup> Docker is an OS project that automates the deployment of applications inside software containers. Azure cloud technology promotes Docker in order to enable users to choose the tools and solution that best suit your needs for Docker container orchestration and scale operations. Source: <https://azure.microsoft.com/en-us/services/container-service/>

<sup>34</sup> On March 11, 2017, at 11:40 and some seconds am, the download counter shows a rate of 158,600, 212, where the rate of downloads is a fraction of a second. See, <http://www.videolan.org/vlc/index.html>

Fourth, *network externalities* occur when individuals using the same or shared standard or technology within a space, derive benefits such as compatibility and economies of scope or scale. These benefits are scale dependent, meaning that the greater number of users, the larger the benefits, particularly when it comes to the benefits of standardization that come in lower interaction costs and incremental development. Network externalities are an important kind of benefit that occur in case of non-rivalrous technologies, that is, the case where copies of an innovation do not reduce benefits the same way they would in the case of rivalrous goods, (e.g., patented innovations) where (Ricardian) rents accrue to rarity. In the case where network externalities are important, unbridled copying is desirable. In contrast, where network externalities (or the number of users on a shared platform) are not important, technologies will tend to be protected by patents or other entry barriers (such as trade or technology secrets), where the main benefit to the innovator will be a license fee. This quality of OSS technology platforms is relevant to our discussion because if an OSS project (or a project's technology) creates 'positive network externalities', then there will be more incentives for its developers to increase its ubiquity (i.e., turning it into the dominant industry standard).

Fifth and last, OSS technology is *more process oriented* than product oriented. The implication of this attribute is twofold. First, openness is central to OSS collaborative RDIP as it enables inclusion of more able developers in its development process. Second, if openness and the open collaborative process not managed properly and the *synthesis and synergy of idiosyncratic assets* are not duly accomplished then the project outcome is of no value to marketplace.

These five qualities—or as I call them inherent particularities of OSS technology—build the case for researchers of open and distributed innovation process to view OSS as a unique phenomenon that needs a theory development effort having two major foci. First, we need to take into consideration the inherent particularities of the OSS

technology; and, how they may unfold in its collaborative RDIP. Second, we have to be observant and vigilant not to follow previously developed paradigm religiously without considering their explanatory power concerning this particular technology.

### 3.5 Theoretical perspectives on OSS development process as a technological innovation

*“Breaking with many established assumptions about how innovation ought to work, open source software projects offer eye-opening examples of novel innovation practices for students and practitioners in many fields.”*

(von Krogh & von Hippel, 2006, p.975)

OSS is a software product that is created collectively and made available online gratis for potential users. The creation and innovation process are open to all potential capable contributors and any software developer (either an individual or a firm) who can voluntarily collaborate with others to develop the software for fun, to satisfy their personal or organizational needs, to gain experience, or to demonstrate their capabilities to attract potential employers. The OSS projects are posted and maintained through internet-based communities of software developers and anyone who is interested and has the expertise can easily start a new OSS project. As OSS development process involves a mix of social, economic, and innovation related factors it has increasingly become under research spotlight being perceived as an exotic test ground for investigating open and distributed innovation and/or “peer production” (e.g., see Feller et al., 2008; Benkler, 2006, 2002).

Thus far, OSS has spurred academic research in a few key areas. Several authors have focused their attention on the “*motivations*” which induce behavior in users and developers and cause their contributions to OSS projects (e.g., Lakhani & Wolf, 2005; Lerner & Tirole, 2002). Others (e.g., Schaarschmidt et al., 2015; O’Mahony &



Ferraro, 2007; Shah, 2006; Kogut & Metiu, 2001) have been intrigued by “*governance*” issues within OSS communities. Some other researchers (e.g., West & O’Mahony, 2005; Scacchi 2002) have found the “*organization*” of OSS projects and communities of importance and devoted their efforts to explore the topic. Yet a few researchers (e.g., von Hippel, 2005; von Krogh et al., 2005; Feller & Fitzgerald, 2000) have made an effort to explore and unbox the unorthodox “*innovation process*” of OSS development.

In what follows, I will mention three core issues concerning OSS R&D and innovation process (RDIP) that have given rise to the prominence to OSS phenomenon and spurred scholars to approach the subject from different theoretical perspectives. These three core issues are: 1) Motivation to engage in OSS development process; 2) measures of OSS success; and 3) governance, organization, and innovation process of OSS development. Each section reviews the related literature with the aim of developing the discussion to highlight relevant problem areas.

### 3.5.1 Theories about why developers engage in OSS development process

Rapid diffusion of OSS, significant investments made by corporations in OSS projects, and the new organization structure (collaborative software development) are among the three important factors that have given rise to the prominence of OSS phenomenon (Lerner & Tirole, 2002). Yet, none of these factors explains *why individuals and firms contribute to OSS Development*. Therefore, ‘Motivation’ to contribute to OSS development process as a technological innovation has been always a core issue while researching OSS (see, e.g., Roberts et al., 2006; Shah, 2006; von Krogh & von Hippel, 2003; Lerner & Tirole, 2002).

The very basic question with regards to motivation to invest time, resources and efforts has been originally framed by Glass (1999, p. 104): “*I don’t know who these crazy people are who want to write, read and even revise all that code without being paid anything for it at all*”. Similarly, Lerner and Tirole (2002, p. 198) are intrigued by “*Why should thousands of top-notch programmers contribute freely to the provision of a public good?*” In view of this line of query, several authors have attempted to approach OSS projects and their developers to provide answers and explanations in form of plausible theories (see, e.g., Bitzer et al., 2007; Bonaccorsi & Rossi, 2006; Shah, 2006; Roberts et al., 2006; Lerner et al., 2006; von Hippel & von Krogh, 2003; Bonaccorsi & Rossi, 2003; Hertel et al., 2003; Lerner & Tirole, 2002).

Bitzer et al., (2004; as cited in Bonaccorsi & Rossi, 2006) assert that firms contribute to OSS projects to *profit* while individuals’ participation seems to be encouraged by a *mix of intrinsic and extrinsic* motives. Bonaccorsi & Rossi (2006) further stress the important role of social and ideological motives.

There are three broad classes of incentives: *economic, social* and *technological*. In addition, there is distinction between micro and macro levels of motivation where individuals refers to the former and organizations to the latter (See pioneering work by Feller & Fitzgerald, 2000).

Lerner and Tirole (2002) focus on *programmers’* contribution to code development. They mention that a programmer is mainly motivated to participate if she receives a net benefit out of her activity; meaning that she engages in a “cost-benefit analysis”. The costs can be the monetary or opportunity cost; and the benefits can include learning or improved performance, enjoying doing something fun, or delayed career-related incentives.

For example, as mentioned by Lerner and Tirole (2002, p. 213), “A ‘cool’ OS project may be more fun than a routine task”. The career related incentives can concern: “future job offers”, “shares in commercial open source-based companies” or “future access to the venture capital market”. Further, Lerner and Tirole (2002) highlight “ego gratification incentive” as a motivation which is rooted in a programmer’s desire for peer recognition. All in all, these incentives can be explained by the “signaling incentive” (for further information see, Holmstrom, 1999); an economic theory that suggests: a) “the more visible the performance to the relevant audience (peers, labor market, venture capital community)”; b) “the higher the impact of effort on performance”; and c) “the more informative the performance about talent” (Lerner & Tirole, 2002, p. 214).

As far as commercial firms are considered, the literature shows that the private sector is reluctant in mimicking the “visibility performance reached in the open source world” (Lerner & Tirole, 2002, p. 223) mainly out of fear of losing their employees to competitors (Lerner & Tirole, 2002) or losing their trade secrets (Ronde, 1999). Therefore, habits of OS world such as “promotion of widespread code sharing” become limited to confinement of a single corporation at best which still is fraught with the hurdles imposed by “existing organizational forms” (Lerner & Tirole, 2002).

However, irrespective of challenges of adopting OS way of software development, commercial firms<sup>35</sup> have developed strategies to benefit from OSS development methodology. Lerner and Tirole (2002) mention three main strategies used by commercial firms to enjoy OSS commons. These are: 1) *Living symbiotically off an OS project*; 2) *Code release*; and 3) *Intermediaries*.

---

<sup>35</sup> The literature uses the term commercial firms or corporations to refer to private software firms which produce proprietary software products, e.g., Microsoft, IBM, Oracle, etc.

The first strategy refers to firms' commercially providing complementary services and products that are not supplied efficiently by the OS community (Lerner & Tirole, 2002). These firms are mainly known as OS or OSS firms. Perhaps the most well-known one is Red Hat® Inc. which is perceived as the world's OS leader. As their primary motivation is to gain profit by providing complementary services, these firms allocate their human capital (programmers) to contribute to OSS projects they are involved with (Lerner & Tirole, 2002). In addition, considering the fact that these firms cannot appropriate all the benefits of their investments in OSS projects, "free-rider problem" is associated with their efforts to connect with communities (Lerner & Tirole, 2002). The second strategy, which is a more proactive in nature, involves releasing the source code of existing proprietary software in an attempt to create an OS project which can attract voluntary contributors (Lerner & Tirole, 2002). This strategy also involves putting in place a proper governance structure (Lerner & Tirole, 2002). Under different conditions, firms may be motivated to opt for this strategy. For example, firm may gain profits due to sales of proprietary 'complementary' segment which can offset any profits that would have been made in the primary segment, had it not been converted to open source (Lerner & Tirole, 2002). Lastly, "Intermediaries" refers to "the creation of organizations such as Collab.Net [sic] as efforts to certify corporate open source development programs, just as investment banks and venture capitalists play a certification role for new firms" (Lerner & Tirole, 2002, p. 228). CollabNet – which functions as an enterprise funded by VC in 1999 – organizes OS projects for private firms (e.g., Philips, Intel, HP, etc.) that intend to develop part of their software through OS methodology. For example, one of the first products of CollabNet was "SourceXchange", a software development marketplace, where developers could bid for contracts to develop OSS applications posted by corporations (Wikipedia, 2014)<sup>36</sup>.

---

<sup>36</sup> For further information on CollabNet please refer to the Wikipedia page at: [http://en.wikipedia.org/wiki/CollabNet#cite\\_note-3](http://en.wikipedia.org/wiki/CollabNet#cite_note-3)

Bonaccorsi and Rossi (2006) emphasize the incentives of firms adopting OS business models. They describe *economic* motivations of individuals as the classical “cost-benefit framework” of economic theory where the cost of participation in an OS project is very marginal for an individual participant. Similarly, Lakhani and von Hippel (2003, p. 939)’s analysis of execution of “mundane but necessary” tasks in the field support of Apache web server software reveal that “the actual answering of questions... took up only 2% of an information provider’s time on site, with providers reporting that they invested only 1–5 min per question answered.”

However, this marginal time (represents marginal cost for contributing) spent on answering questions (a form of contribution to OSS community) was also because the information providers only shared information they already knew “off the shelf”. They seldom did new problem solving or searching in order to provide additional information to a help-seeker (Lakhani & von Hippel, p. 939). As far as the benefit accrued to participants is concerned, Lakhani and von Hippel (2003, p. 923) find that “98% of the effort expended by information providers in fact returns direct learning benefits to those providers”.

*Social* motivations, on the other hand, are concerned with individuals’ motivation to gain social status, peer recognition, and reputation (see, Bonaccorsi & Rossi, 2006; Zeitlyn, 2003; Raymond, 1999). Raymond (1999), for example, views OS development from a “gift culture” perspective. Zeitlyn (2003, p. 1287) notes that “a gift includes an obligation to make a return presentation. This compulsion to return a gift has special force in a small social world. The public world of the net, especially that of the software engineer, is a *very* small world no matter where they are physically based”. In fact, gift cultures are based on “gift economies”, in which social relations are not regulated by the possession or exchange of money or commodities; but they are known by the creation and maintenance of social relationships based on the economy of gift exchange (Bergquist & Ljungberg, 2001, p. 308). The same goes

true with OSS programmers where successful gift giving can lead to gaining reputation and social status. Bergquist and Ljungberg (2001) analysis of gift giving practices within OS community reveals three main purposes of such behavior: 1) creates openness; 2) shapes relationships among community members by awarding the givers a higher status among their peers; and 3) circulating the new ideas.

*Technological* motivations are enabled by the very nature of OSS which involves open access to the source code and transparent development process. In fact, OSSD process provides an excellent opportunity to those individuals who wish to dive into the source code, study it, learn from it, and use it to resolve new issues (Bonaccorsi & Rossi, 2006). More importantly, the interaction and the iterative peer-review process in code development and refinement lead to learning opportunities which are not available to free-riders (von Hippel & von Krogh, 2003). The mentoring that is being offered to those volunteers who go deep into iterative peer-reviewed coding is among the benefits which are particularly exclusive to those intensely involved. Such involvement in open coding builds necessary software development competencies that cannot be obtained through working on proprietary software projects in contrived environments. Furthermore, other scholars view open code writing as motivated by *tapping an unmet market need* (e.g., Feller & Fitzgerald, 2000) or “an itch worth scratching” (Raymond, 1999). The story of the development of Perl language by a systems administrator to satisfy an unmet need is an attestation to technological motivations.

OSS firms’ motivations to take part in OSSD are also classified into the three categories by Bonaccorsi and Rossi (2006). The main *economic motivation* for OS firms to engage in OS movement is to escape the financial burden of paying license fees to proprietary software developers like Microsoft or Oracle (Bonaccorsi & Rossi, 2006). Therefore, OSS firms develop different business models (see Perr et al., 2010 for a typology of these business models) which are centered mainly on offering

complementary OSS-related services. Also, as discussed in the previous chapter (see section 2.5.7 Does OSS business models inform us about OBM concept?) OSS firms develop different types of relationships with community of OSS developers in order to benefit from the collaboratively developed technology to different degrees (see for details: Dahlander & Magnusson, 2008; Dahlander & Magnusson, 2005). Mainstream software firms like IBM also benefit economically from OSS communities by releasing their proprietary source code into the OS domain through: exploiting the contributions of individual developers, benefiting from the resultant R&D spillovers, and selling OSS-based products (Bonaccorsi & Rossi, 2006). Lastly, software firms use their engagement with OSS communities as medium to attract talented developers for hiring purposes (Wichmann, 2002; as cited in Bonaccorsi & Rossi, 2006).

Firms also have *social motivations* for their involvement in OSS movement and building a relationship with OSS communities (see Bonaccorsi & Rossi, 2006; Dahlander & Magnusson, 2005). As a case in point, those firms that use OS code but do not conform to the non-written rules of OS community face considerable competitive disadvantages (Osterloh et al., 2002; as cited in Bonaccorsi & Rossi, 2006). Behaviors such as turning open code into proprietary pieces are in contrast with the social norms governing the OSS community projects and are therefore viewed as betrayal over the trust of the individual developers (Bonaccorsi & Rossi, 2006, p. 49). Similarly, as Dahlander and Magnusson (2005, p. 487-488)'s typology of firm-community relationships shows, firms that resort to a "parasitic approach"<sup>37</sup> towards community of OSS are "perceived as a negative influence by the community, either in terms of its violation of basic norms, values and principles, or that it is simply perceived as a free rider". Therefore, part of social motivations for firms includes collaboration with and contribution to OSS projects to maintain either a

---

<sup>37</sup> The *parasitic approach* implies that the firm only focuses on its own benefits, without taking into account that its actions might harm the community (Dahlander & Magnusson, 2005, p. 487).



“symbiotic” or “commensalistic” relationship with OSS communities. The symbiotic approach leads to a situation where the firm co-develops hand in hand with the community; and the commensalistic approach captures an intermediate way to inter-relate to the community and eventually to benefit from the co-existence with another entity while leaving it without harm (Dahlander & Magnusson, 2005).

Lastly, technologically, firms are motivated to engage in a relationship with OSS communities because they can receive feedback and contributions from the OSS communities; and these technical benefits in turn lowers their R&D costs and improves the quality of the software they produce (Bonaccorsi & Rossi, 2006).

Shah (2006)’s study also provides an interesting perspective on motivations of individual participants concerning their engagement in OSS projects. She makes distinctions between open and gated communities; as well as creation of code vs. contribution to code development. Shah (2006, p. 1004) divides individual participants into two categories: *need-driven* participants and *hobbyist* participants.

Need-driven participants are motivated by their needs (e.g., “the need for software-related changes”). An individual has a need and therefore cannot wait for others to get the job done and solve the pressing problem. However, having created the code, participants make contributions to code development due to some other reasons such as “reciprocity, future improvements, source code commits, and career concerns” (Shah, 2006, p. 1005). Hobbyists’ participation, on the other hand, creates codes as they seek fun and enjoyment in the process of coding (Shah, 2006). Need-driven participants and hobbyists are complementary to each other therefore forging a symbiotic relationship. The useful challenges hobbyists are looking for can be indeed the needs of need-driven participants that are left unattended in the community.



Roberts et al. (2006) develop a theoretical model that explains the relationship between motivations (intrinsic and extrinsic), participation and performance of OSS developers. Authors further test their proposed model through a data set gathered from longitudinal research on software developers from the Apache projects. They discover that the developers' motivations are not independent but rather they are related in complex ways. They shed light on the interplay between intrinsic and extrinsic motivations. For example, they do not find evidence to support diminishing intrinsic motivation (use-value) while extrinsic motivations (paid contribution) are introduced. Further, they find that motivations types influence participation of individuals differently. Lastly, the contribution levels of individuals have a positive bearing on their performance rankings in the context of the projects they contribute to them. Table 3.1 provides a summary of motivations mentioned in this section. As Table 3.1 shows, past research has mainly focused on identifying and discussing motivations of individuals and software firms. The gap, thus, remains to explore the OSS RDIP in order to find out whether there are other sorts of contributors to OSS development process. If so, what could be their motivations?

Table 3.1. Major Motivations Underlying Contributions to OSS Projects

Level of Analysis	Motivation	Motivation Class	Underpinning theory	Reference
Individual	Direct learning benefits	Technological	–	Lakhani and von Hippel (2003); Lakhani and Wolf (2005); Hertel et al. (2003)
	Fun, Enjoy, doing a cool task; hobby	Personal and social (intrinsic)	–	Ghosh (1998); Lerner and Tirole (2002); Constant et al. (1996); Torvalds and Diamond (2001); Hertel et al. (2003); Shah (2006)
	Direct utility or own use (increasing own technical performance at workplace)	Technological (intrinsic and extrinsic)	–	Lerner and Tirole (2002)
	Gaining future career benefits	Economic (extrinsic)	Signaling incentive (Holmstrom, 1999)	Lerner and Tirole (2002); Hertel et al. (2003)
	The ego gratification incentive (desire for peer recognition); reputation	Personal and social (intrinsic)	Signaling incentive (Holmstrom, 1999)	Lerner and Tirole (2002); Lakhani and Wolf (2005)
	Competition with other developers (either within OSS projects or between OSS projects and commercial software projects)	Personal and social (intrinsic)	–	Bezroukov (1999a; as cited in Hertel et al., 2003, p. 1162); Hertel et al. (2003)
	Being identified with community (sense of belonging)	Personal and social (intrinsic)	Extended Klandermans Model of voluntary action in social movements (Klandermans, 1997; Simon et al., 1998).	Hertel et al. (2003)

Commercial software firm and/or OSS firm

Norm-oriented motives related to reactions of relevant others (family, friends, colleagues)	Personal and social (intrinsic)	Extended Klandermans Model of voluntary action in social movements (Klandermans, 1997)	Hertel et al. (2003)
“commercially providing complementary services [as installation, integration, maintenance, and support] and products that are not supplied efficiently by the open source community”	Economic (appropriate benefits through spillover effects)	Spillover effect; cost-benefit analysis, lowering production costs by not paying royalties	Lerner and Tirole (2002, p. 224); Lerner et al. (2006); Wichmann (2002); Bonacorssi et al. (2006)
Build public relations with programmers and customers	Economic (appropriate benefits through spillover effects)	Spillover effect	Lerner and Tirole (2002; 2005); Lerner et al. (2006)
Respect the psychological implicit contract between OSS firm and programmers	Social	Conformity to social norms induces cooperative behavior	Osterloh et al. (2002; as cited in Bonacorssi & Rossi, 2006)
Company programmers learn from their participation and benefit the company	- Economic (Lowering costs of R&D); - Technological (appropriate benefits through spillover effects, <i>learning</i> , which improve software quality)	Production cost-benefit analysis; Spillover effect	Lerner and Tirole (2002); Lerner et al. (2006)
Access to unique community-based resources	Economic and technological	Resource-based view	Dahlander and Wallin (2006)
Gain indirect revenues through selling open source related products	Economic	Cost-benefit analysis, lowering production costs by not paying royalties	
For hiring purposes	Economic	Resource-based view	Wichmann (2002)

Source: Compiled by the author

### 3.5.2 The commercialization challenge of OSS

The landscape of OSS development has undergone a transformation, such that it has turned into “a more mainstream and commercially viable form” (Fitzgerald, 2006, p. 587). Such metamorphosis has affected all phases of development process including planning, analysis, design, and implementation, to the extent that it has turned into a strategic product (Fitzgerald, 2006). Therefore, issues like “rigorous project management”, “branding” of OSS products, offering a “whole-product concept of a market-driven business approach” have become integrated into the OSS phenomenon of today’s market place (see, Fitzgerald, 2006). All these factors collectively contribute to development of professional OS products and services, increasing trustworthiness of the technologies, gaining leadership positions in target markets, and delivering solutions that actually address customers’ needs.

Others (Feller et al., 2008, p. 476), on the other hand, have been rather agnostic about effectiveness of OSS in addressing real market needs. They emphasize, “the community-based peer production of software often falls short of creating customized software products in the sense that individual and corporate consumers understand”. Woods and Guliani (2005) underline this shortcoming, describing it as the need to “productize” OS service in order to bridge the gap between private enterprises and individual end users.

A review of the past research on the commercialization process of OSS also shows that authors have mainly focused on how very *large* (software) corporations such as IBM, Google, Sun<sup>38</sup>, or OSS firms that invest in communities of OSS developers, collaboratively create value, and address the challenges of appropriation (e.g., West & Gallagher, 2006; Dahlander & Magnusson, 2008). A good part of this literature has also focused on different types of business models and their role in making the relationship between OSS firms and communities work in a profitable manner (e.g., Dahlander & Magnusson, 2005; Dahlander, 2005; Perr et al., 2010).

<sup>38</sup> On January 27, 2010, Sun Microsystems, Inc. was acquired by Oracle Corporation. (Source: [https://en.wikipedia.org/wiki/Sun\\_Microsystems](https://en.wikipedia.org/wiki/Sun_Microsystems))

However, as noted by Feller et al. (2008) and Fitzgerald (2006) the process of commercializing and productizing OSS oftentimes extend beyond the capabilities of many OSS firms; thereby creating the need for “cooperative business networks”.

It is true that “commercial open source” is software that a for-profit entity develops and controls its development direction. Naturally, the firm behind the commercial OSS has certain degree of power to determine what gets accepted and integrated into the software source code and what will be seriously discussed for the next implementation such as the case of MySQL and its MySQL database<sup>39</sup> (Riehle, 2007). However, services to be provided on strategic and commercial OSS products are not limited to the founders of OSS project. For example, many OSS firms provide services on Odoo<sup>40</sup>, and they are also involved in the innovation process through developing new features (modules), collaborating together in development process and maintaining the features over time. In fact, OSS industry is not a saturated marketplace; there is always a need for new capable firms to enter the competition.

These conflicting views on OSS as a commercially viable and competitive product coupled with the need to create cooperative business networks to address commercialization challenge make a case to further investigate the OSS R&D and innovation process (RDIP) in order to discover factors that affect effective commercialization and further sustainability of OSS products and services. For instance, knowing about the consequences of *not* engaging in collaborative RDIP can better inform us of what can actually contribute to successful OSS commercialization.

### 3.5.3 Understanding measures of ‘success’ in context of OSS

‘Success’ of OSS project is a multi-faceted and fuzzy concept which depends on many factors and can be viewed through different perspectives. In fact, understanding the

---

<sup>39</sup> Available at [www.mysql.com](http://www.mysql.com)

<sup>40</sup> Formerly known as OpenERP— a very successful OS enterprise resource planning software

predictors of OSS success is one of the key areas of interest (e.g., Subramaniam et al., 2009).

One of the startling facts about OSS phenomenon is that today thousands of OSS projects exist, filling a wide range of individuals' and organizations' need gaps. Table 3.2, below, demonstrates the basic statistics concerning three major websites that provide an online platform for OSS projects and the interactions among their contributors and users.

Table 3.2. Basic Statistics on Major Online Community Platforms

Host	Number of projects or repositories being supported	Lines of codes in total	Registered users	No. of contributors	Downloads per day	Last update on statistics
Sourceforge .net	430,000	N.A.	3.7 m	N.A.	4,800,000	February 10, 2015
Black Duck Open Hub (formerly Ohloh.net)	668,055	31,042,267 ,464	N.A.	3,695,026	N.A.	February 10, 2015
GitHub	20.1 m repositories	N.A.	3.4 m	8.6 m	N.A.	February 10, 2015

*Source:* Compiled by the author based on available data online

Note 1: Numbers are constantly changing and they should be taken as approximations

Note 2: N.A.: Not Available

Among most successful OSS projects are GNU/Linux (computer operating system), Mozilla web browser, Apache (server software), PHP and Perl (programming language); OpenStack (cloud software), Joomla! And WordPress (web content management), SugarCRM (customer relationship management), and MySQL database, to name a few.



However, irrespective of highly successful projects, “the number of failed or dormant OSS projects is also notable” (Lee et al., 2009, p. 426). Based on statistics provided by the one of the most popular OS portals (SourceForge: (<http://sourceforge.net/>), most OSS projects have failed: 58% of the projects do not move beyond the alpha developmental stage, 22% of them remain in the planning phase, 17% remain in the pre-alpha phase, and some become inactive (Lee et al., 2009, p. 426). Similarly, a World Bank study has reported the failure rate of more than 50% for OSS projects (as cited in Lee et al., 2009, p. 426). Most recently, based on online news by Donnie Berkholz, a senior analyst at RedMonk, it is reported that “the vast majority of repositories (>98%) are modified only in the year they are created, and they’re never touched again” (Berkholz, 2014). This suggests that more than 98% of all projects on GitHub do not see any development beyond the first year they have been written (see, Asay, 2014).

Unconventional to the perceived wisdom, some practitioners celebrate OSS success by mentioning the magnitude and activity level of these major online community platforms as an attestation to the success and growth of OSS as a technological innovation. They believe, “open source failure is its greatest success” (Asay, 2014). They view OS as all about ‘experimentation’ and ‘iteration’, which is why a 98% failure rate may be well the best sign of its success (see, Asay, 2014). Instead of focusing on the failure rate, which has remained so dramatic in the past decades, they shift the focus on the “volume and breath” of the OS projects, which indeed has skyrocketed. In the article: “Open source failure is its greatest success” by Matt Asay (2014), he mentions that “every year (plus a month or so), both the total number of projects and lines of code double”. This growth in the “overall number of projects” and “magnitude of lines of codes” is viewed as sign of “tremendous success”, simply because they shed light on the ‘evolution’ and ‘evolutionary nature’ of OS projects where failure is seen as the “norm”. Thus, by “redefining success” (taking a holistic view of OS phenomenon) “failure” becomes the greatest “beauty” of OS world. This approach to success, which emphasizes “spirit of iteration” and “experimentation”, resonates well with the Linus’s Law as put forward by Raymond (1999): “*given enough*

*eyeballs, all bugs are shallow*". This means that "given a large enough beta-tester and co-developer base, almost every problem will be characterized quickly and the fix will be obvious to someone" (Raymond, 1999).

On a broad note, this perspective on success, emphasizing experimentation, sits well with notion of "open experiments" in value creation process of open business models (Chesbrough, 2007b). Therefore, those projects that cannot create enough or useful value for users shall not sustain long-term. Several scholars (von Krogh & von Hippel, 2006, p. 975) have called for need to properly understand how OSS functions so that we can examine the well-known negative side effects on freedom to innovate.

However, divergingly yet creatively, others have taken a 'different approach' towards "success" of OSS projects. Having perceived OSS as a form of system development, IS researchers (e.g., Sen et al., 2012; Subramaniam et al., 2009; Lee et al., 2009; Crowston et al., 2006) have approached operationalization of the concept by building upon "success measures" identified in the IS literature (e.g., DeLone & McLean, 2003, 2002, 1992; Rai et al. 2002; Seddon et al. 1999; Seddon 1997).

Measuring the success of OSS projects is important for two reasons (Crowston et al., 2006, p. 123). First, to have a better understanding of determinants of success can help project leaders in assessing their projects. For example, in projects that involve third party sponsors, the investors are keen on knowing about the return on their investment (ROI). Second, to learn from OSS development process, particularly from "teams that are working well we need to have a definition of 'working well'" (Crowston et al., 2006, p. 123). This second factor is based on the premise that OSS development is "an increasingly visible and copied mode of systems development" (Ibid., p. 123). Further, many users including individuals and corporations are dependent on OSS products such as Linux (Crowston et al., 2006). Yet, we still know little about "how people in these communities coordinate software development across different settings" or "what software processes, work practices, and organizational contexts are necessary to their success" (Scacchi, 2002). More importantly, literature (Ghosh, 2002; as cited in



Crowston et al., 2006) has emphasized the need to learn from OS modes of organization and production with the aim of possibly applying the insights to other areas as well as to better concert the efforts of OS projects.

Irrespective of importance of why project success should be studied, IS researchers have identified 'success factors' differently and measured their impacts through different methodologies.

Measures of success are different and varied. In crudest form, as OSS projects depend on input added by volunteers, the ability of a project to attract the interest of developers as well as contributions from them is a key success measure (Stewart et al., 2006). Such interest can be measured through the traffic on the OSS projects website and the number of downloads of the software code (Crowston et al., 2003). Stewart et al. (2006) also identify interest over time (i.e., change in the number of subscribers) and amount of development activity (i.e. the number of files released) as measures of OSS project success.

Several scholars (Crowston et al., 2006; Stewart & Gosain, 2006) have proposed measures such as project activity levels, release of new features, development team/community size (i.e., the number of active contributors to the project), and the time taken to fix software bugs. Subramaniam et al. (2009) use developer interest, user interest, and project activity as measures of project success and investigate the interaction effects among them.

Other researchers have emphasized the role of developers and their voluntary contributions concerning creating, debugging, and maintaining the OSS project as critical to OSS success. Therefore, they have viewed developer participation as critical to project success and have focused on their motivations, incentives and interests (e.g., Bonaccorsi & Rossi, 2003; Lerner & Tirole, 2002).

More recently, Sen et al. (2012) measure OSS success through the 'number of subscribers' and 'developers' working on an OSS project. Sen et al. (2012, p. 365) emphasize "subscriber base" as a key measure for success, which has been ignored by previous studies (e.g., Subramaniam et al., 2009). They define *subscriber base* as "the number of individuals who chose to receive regular information (e.g., project progress, new release information, any major changes to the project team etc.) on the project" (Sen et al. 2012, p. 365).

This determinant of success is important for several reasons. First, growth in number of subscribers shows the increasing interest in the project from committed and potential users, competitors, and other stakeholders (Sen et al., 2012). Second, subscribers have chosen to be part of the project development and therefore they are received as people with "a deeper interest in the OSS project" (Ibid., p. 365). Consequently, information about such a large group of loyal users can be useful to project management team. Lastly, Sen et al. (2012, p. 365) mention that subscribers can provide "clearer and better directions with respect to project development for long-term success". They are also perceived as "committed users" who are also more likely to support the project financially if needed because of their higher interest (Sen et al., 2012).

The review of the literature on measures of success of OSS projects shows the multi-dimensional nature of OSS success. As OSS projects are targeted at different types of audiences or users (Sen et al., 2012, p. 366), different types of success measures may become more or less important in the role they play. OSS is not just about developing or modifying software; increasingly more non-developer users are relying on OSS for personal and business needs (Lerner & Tirole, 2005). For example, clients of OSS firms are amongst the most important users as well as audiences of OSS projects so much so that without them, the whole sector of OS or OSS firms falls apart. They are the ones that sponsor commercial features of OSS projects and keep the core development team on payrolls. Especially, as we discussed earlier, OSS has been transforming into a commercially viable option (Fitzgerald, 2006) which competes on certain fronts with proprietary software products. Yet we do not have enough information about the

development processes for commercial OSS products (Aksoy-Yurdagul, 2015). Equally important is the role of clients in successfully developing strategic OSS which is built upon collective effort but it responds to different target groups' technological needs. Therefore, the clients can be considered as active participants who can influence the success of OSS projects for a number of reasons.

First, like subscribers to OSS projects as developers or users, clients are enterprises that are considered as users and beneficiaries of OSS. The efficiency and effectiveness of their information systems (ISs) rely, to different degrees, on OSS tools, applications, and support services which are provided by communities of OSS and private OSS firms. Clients' innovation performance and manufacturing success are also depending, to varying degrees, on their supporting ISs which include open source applications. Therefore, they have 'incentive' to be part of the development team and consequently be influential in the success of OSS project. The influential role of interest shown by users of OSS has been perceived as an important indicator of project success (Stewart et al., 2006), yet it has not been explored in the context of clients of OSS firms.

Second, OSS success heavily relies on developers: their contributions in creating codes, improving them through bug fixes, and maintaining the code base in the long run through active participation (these are also measures of success as discussed above). Therefore, clients which have resources such as knowledgeable and experienced IT personnel, testing infrastructures, target market knowledge and feedback, etc. can play a significant role by their contributions to the OSS development process; and thereby influence OSS success.

On a more detailed note, Grewal et al. (2006) mention that the criteria for success of OSS projects should encompass both the technical achievements of a project, as well as indicators of market or commercial success: i.e., technical and commercial success. Such classification is in line with IS' literature on software success (e.g., Rai et al., 2002) as well as the literature on R&D success in new product development (NPD) dating back to Mansfield and Wagner (1975).

As Weber (2004, p. 77) asserts, “Prediction is actually the enemy in software testing”. Therefore, instead of guessing and predicting how a piece of software functions in different situations or what functionalities must be added to an open source application, we need to consult the end-users and design software from their perspective. Thus far, to the best of my knowledge, the role of non-software enterprise firms or clients as developers of OSS (projects) and their impact on their success have remained unexplored. Weber (2004, p. 216) has also highlighted this gap. He further claims, “the role of the ‘customer’ in the production process, specifically as it plays out within business models” is a central conceptual question to be further studied in the context of OSS development. In line with this argument and based on the reviewed literature on OSS success, we can view enterprise users of OSS *as stakeholders of OSS projects*. *Thus, it is safe to argue that their interest can influence the OSS project commercial and technical success*. Such conceptualization further motivates me to find more about the way (‘how’) non-software enterprise firms engage in OSS technological collaborations.

#### 3.5.4 OSS innovation process

If there is one thing that surprises those who are not well acquainted with OSS development process, it is the level of heterogeneity involved with regards to types of actors and their motivations to participate in its joint development process. This diversity and the difficulty associated with managing it put the spotlight on issues of project organization and governance within the innovation process. In what follows, I review the most notable works on these issues.

First, OSS “innovation process” defies the conventional assumptions that underpin and promote innovation and necessitate the intellectual property system (e.g., von Hippel, 2005; Dahlander & Magnusson, 2005, Ulhøi, 2004). A few scholars (Ulhøi, 2004; Tuomi, 2002) have also viewed OSS projects as creating an institutional alternative to ‘firm-based’ innovation, so much so that Lerner et al. (2006, p. 114) claim that “there

are substantial differences between open-source projects and traditional innovative efforts in private firms”.

Historically, based on the doctrine of ‘private investment model’, innovation has been mainly supported by private investment paradigm where the investor collects private returns from his investment (Demsetz, 1967). Based on this view, “copyright and patent law provide a sound basis for an economically efficient system of protection... [focus is on software industry]” (Dam, 1995, p.321). Therefore, one’s innovation cannot be integrated into the ‘commons’ because if this happens the innovator will incur losses due to hazard of free-riding which eventually destroys the innovator’s incentive to further innovate (see, Granstrand, 1999; Dam, 1995; Demsetz, 1967). Obviously, it also negatively affects the sustainability of innovation process long-term, as innovation is more contingent upon continued investment in the R&D and other innovation-related activities.

However, in face of success stories of OSS projects and its technological collaborations, the conventional perception of how innovation process can effectively function has changed its course. For example, today, thousands of OSS projects exist addressing a wide range of individuals’ and organizations’ needs (as shown in Table 3.2)

In view of success and growth of OSS projects (e.g., GNU/Linux, Apache, Perl, OpenStack, etc.), several scholars (von Krogh & von Hippel, 2006, p. 975) call for the need to take a *fresh perspective* (“fresh eyes”) to properly understand how OSS functions in order to examine the negative side effects of freedom to innovate and adopt (e.g., Heller & Eisenberg, 1998)<sup>41</sup>.

Additionally, OSS is a “public good” or “commons”. It demonstrates attributes of *non-exclusivity* and *non-rivalry*. An individual or a firm can download, use, and share OSS tools and solutions without excluding others from benefiting from the innovation; and

---

<sup>41</sup> “fresh eyes” refers to the private-collective innovation model proposed by von Hippel & von Krogh (2003).

where one individual or firm uses an OSS product, they do not reduce its availability and use to others (see, e.g., von Hippel & von Krogh, 2003; O'Mahony, 2003). *However*, OSS also differs from a pure public good as it involves a copyright-based license to keep private intellectual property claims out of the way of both software innovators and software adopters while at the same time preserving a commons of software code that everyone can access (O'Mahony, 2003). Furthermore, OSS projects resort to use of "legal and normative tactics to protect their source code from proprietary appropriation and to protect their collective identity and reputation" (O'Mahony, 2003, p. 1194).

Therefore, having qualities of both public and private goods as well as being produced under private-collective innovation model highlight OSS as a unique phenomenon to further investigate the concept of openness, and open and distributed innovation within software industry.

Earlier attempts by scholars have provided us with 'conceptual frameworks' for understanding the unique character of and the specific requirements for OSS innovation process (Scacchi, 2002; Feller & Fitzgerald, 2000).

For instance, Scacchi (2002) describes, examines and compares the OSS development processes within four communities and identify eight kinds of software "informalisms" which play a pivotal role in the elicitation, analysis, specification, validation and management of requirements for developing OSS systems. Mockus et al. (2002) examine data from two major OS projects (the Apache web server and the Mozilla browser) in order to further clarify OSS development and maintenance as claimed to be competing with traditional commercial software development method. Mockus et al. (2002) quantify several issues such as aspects of developer participation, core team size, code ownership, productivity, defect density, and problem resolution intervals for these two OSS projects in order to develop several hypotheses by comparing the Apache project with several commercial projects. Von Krogh et al. (2003) looks into the OSS innovation process by focusing on the creation of the project named "Freenet"

adopting an inductive theory building approach. They explore the “strategies and processes” by which new people join the existing community of software developers, and how they initially contribute code. Based on their analysis of data, the authors generate four constructs: “joining script”, “specialization”, “contribution barriers”, and “feature gifts”, and propose relationships among them. Their research sheds light on the evolutionary nature of OSS development process that shows how individuals join the projects and follow a social “joining script”, contribute to development process and undergo a transition from a joiner to a newcomer and eventually a committer.

David and Rullani (2008) adopt a “systems analysis perspective” to analyze the critical properties of OSS mode of innovation. They rely their theoretical perspective on March’s (1991) framework of innovation within a firm. However, they ‘re-scale’ it so as to make it fit the characteristics of innovation system of distributed organization of interacting agents in a virtual collaboration environment (as suggested by OSS innovation system). David and Rullani (2008) view “exploration” and “exploitation” phases (March, 1991) as suitable to be applied to the innovation system of OSS. According to David and Rullani (2008, p. 647) the “innovation system of virtual collaboration environment” mirrors the two processes. Meaning that, *exploration* captures the interactions among agents searching information and knowledge resources to apply in designing new software products; while *exploitation* involves the mobilization of individuals’ capabilities for application of those information and knowledge resources in the software development projects that become established on the platform. Their observation shows that OSS innovation process, as studied through SourceForge platform, tends to be highly “dissipative”; i.e., a very large proportion of the registered developers fail to become even minimally active on the platform. While this could hamper sustainability of the innovation system, a smaller number of the core developers— who persist in creating novelty and the structure of their interactions— may be enough to sustain both the exploration and exploitation phases of the platform’s global dynamics.



While the research on OSS development process clarifies certain realities associated with its innovation process, the issues of commercialization and sustainability of OSS innovation remain elusive.

There are two types of OSS: community and commercial OSS. The former refers to the software that a community develops, rather than a single corporate entity owning the software. Therefore, individual developers, hobbyists and the committers and not necessarily a specific firm (here software firm or OSS firm) make decisions about the software and its direction (e.g., the Apache Web server; Liferay Portal Community Edition<sup>42</sup>). The latter is software that a for-profit entity owns and develops; therefore, the firm or group of firms sponsoring the software development and innovation process hold the copyright and determine what is accepted into the software code base and what to implement next (e.g., MySQL and its MySQL database; Liferay Portal Enterprise Edition License<sup>43</sup>). However, this does not mean that the commercial OSS does not follow the OSS development methodology and its characteristics. There is transparency concerning the social and the technical aspects, but the commercial sponsors have significant decision-making power.

Irrespective of their differences, every OSS project enjoys from and prosper through having more developers joining and contributing to the project. This will help projects evolve into a more user friendly and useful version. They can also diffuse through positive network externality effects (See, e.g., Bonaccorsi & Rossi, 2003). For example, in the client market Microsoft has succeeded in gaining the dominant position with its MS Dos and Windows products, a huge market in which OSS products like LibreOffice, Apache OpenOffice could not largely diffuse.

---

<sup>42</sup> Liferay Portal CE, (LGPL, version 2.1) is the OS version of Liferay's enterprise web platform for building business solutions that deliver immediate results and long-term value. Source: <https://web.liferay.com/community/liferay-projects/liferay-portal/overview>.

<sup>43</sup> In addition to Portal Community Edition ("Portal CE"), Liferay packages and makes available a supported Portal EE, intended for enterprise-grade users and implementations, which is subject to the Liferay End User License Agreement and made available to Liferay Enterprise Subscription customers at no extra charge. Source: <https://www.liferay.com/downloads/ee-license>.



In commercial OSS innovation process, the inclusion of end-users early on in the innovation process and development phase and later in maintenance and further evolution of the project can influence the innovation process positively. They can bring along their own software expertise, real market needs, and tacit knowledge. These can lead to developing software that is more useful and user friendly. In fact, the “potential for economically important software novelty may reside not in the technical modifications of the code *per se*”, it lies in how these modifications can actually “enable the project to explore and possibly fill a new and growing market niche” (David & Rullani, 2008, p. 657).

This view of innovation is consistent with Schumpeter’s thesis that innovation does not necessarily mean introduction of technical novelties, but reapplying the existing product designs and production methods in new market context (Schumpeter, 1934). Involving all enterprise users in the innovation process provides this opportunity for the OSS firms as well as other community developers to enjoy what traditionally is called *external knowledge as a source of diversity*. The OSS ecosystem can exploit these sources of diversity in order to escape the trap of local search and induce wider exploration even when the underlying technology is similar (David & Rullani, 2008, p. 657). It can also lead to exploiting existing technology in new contexts.

In summary, the unique qualities of OSS –resembling both public and private goods, following a private-collective innovation model– provide both opportunities and challenges for its innovation process success and long-term viability. As a non-rival, non-exclusive public good (e.g., community-based projects such as FFmpeg), it can be easily used, exploited, integrated into downstream products and industries and therefore diffused rapidly. This can create network effect and attract more collaborators. However, since it is not purely a private good, managed and controlled by private investors –who collect all returns on their investment– it suffers from abundance of marketing insights and end-user perspectives, thereby effective commercialization. The challenge with OSS innovation process, therefore remains as how to balance these qualities of OSS so as to sustain its innovation process.

### 3.5.5 Governance of OSS projects

OSS development process represents a collective R&D and innovation process (RDIP) reflecting a (social) network phenomenon. The process of technology development includes a network of individuals and firms. For example, in order to better understand the workings of OSS environments, Grewal et al. (2006) have studied the effects of network embeddedness (or the nature of the relationship among projects and developers) on the success of OSS projects. Von Krogh et al. (2003) focus their attention on creation of 'Freenet' (a project aimed at developing a decentralized and anonymous peer-to-peer electronic file sharing network) to find out about the strategies and processes by which new people join the existing community of software developers, and how they initially contribute code.

Research on networks has stated, "Quite literally, networks are reshaping the global business architecture" (Parkhe et al., 2006, p. 560). Parkhe et al. (2006) view networking and networks as important and relevant to management theory and practice due to the 'pervasiveness' and functionality of this phenomenon. In other words, a wide range of "social, digital/electronic, and organizational networks", such the case of OSS development in the community-based model of the OS movement, has defied the conventional constraints of physical geography that puts limits on our communication and collaboration modes and volumes. Therefore, traditional organizational structures are being challenged (and perhaps re-defined) by the networking phenomenon, partly fueled by international competition and advancements in ICT sector.

In some countries (e.g., Japan and South Korea), networks (e.g., intra-firm and inter-firm networks) are of paramount importance so much so that they have become an integral part of the overall economic structure (Gerlach, 1992). Similarly, in some industries (e.g., OSS industry within ICT sector), the networks between OSS firms and communities is an integral part of new product development and R&D process. However, "there is still a considerable discrepancy between the acclamation and attention networks receive and the knowledge we have about the overall functioning of networks" (Provan & Kenis, 2008, p. 229).

Although there are “serendipitous” networks, “goal-directed” networks (Kilduff & Tsai, 2003) are formed by organizations in order to solve problems (e.g., Imperial, 2005; Agranoff & McGuire, 2003). Goal-directed networks are, therefore, set up with a specific purpose, either by those who participate in the network or through mandate and evolve largely through conscious efforts to build coordination (Provan & Kenis, 2008). As the pursuit of a clear goal derives formation of a goal-oriented network, to understand why networks produce the desired outcome, it is critical to understand how a network functions. Therefore, network functioning as “the process by which certain network conditions lead to network outcomes” (Provan & Kenis, 2008, p. 229) becomes a core concept in our investigation. Equally important is the concept of “network effectiveness” which emphasizes why we need networks in the first place. Network effectiveness means, “the attainment of positive network level outcomes that could not normally be achieved by individual organizational participants acting independently” (Provan & Kenis, 2008, p. 230). Network effectiveness also mirrors the concept of synergy, as a collective outcome is greater than the sum of individual efforts of a group of people (see also O’Toole, 1997).

Since networks of OSS development are not legal entities like joint ventures and equity-based alliances, the legal imperative for governance in the sense that it applies to organizational setting does not exist. Nevertheless, that does not mean OSS networks do not need governance to obtain their collective goals. For example, Kogut and Metiu’s (2001) research shows that governance structures of OSS communities play an important role in minimizing the hazard of project forking.

As claimed by Provan and Kenis (2008, p. 231), in the context of goal-directed organizational networks, “some form of governance is necessary to ensure that participants engage in collective and mutually supportive action, that conflict is addressed, and that network resources are acquired and utilized efficiently and effectively”. So is true with OSS networks, as they are goal-directed yet not strictly legal entities, lack formal contracts, and organizational hierarchies. In fact, ‘governance’, the topic that has long been studied by organizational scholars (see, e.g.,

Westphal & Zajac, 1995; Mizruchi, 1983), is believed to have a significant influence on network effectiveness (Provan & Kenis, 2008). This means that if OSS networks are not governed properly, the positive network level outcomes, which are the reasons networks are formed, will not be obtained. Thereby, the collective effort is doomed to fail.

The situation of OSS projects is also peculiar because the OSS innovation process is open to public – perceived by some authors as an outcome of adopting an OI model (Aksoy-Yurdagul, 2015). Furthermore, the OSS technology itself is one with unlimited downstream utilities. Therefore, the level of heterogeneity observed concerning participant types and their incentives is quite higher as compared to, for example, kite surfing communities that only attract those who are into kites and the product being collectively developed has limited utility. Therefore, several authors have studied governance issues concerning OSS projects (see, e.g., von Hippel & von Krogh, 2006).

Kogut and Metiu (2001) endorse the efficiency of OSS development methodology, but highlight the hazard of project forking (i.e., the potential for fragmenting the design into competing versions) as seen in the case of UNIX and Java. The authors claim that “governance structures offer some potential for preventing ‘forking’” (Kogut & Metiu, 2001, p. 257).

O’Mahony (2003) raises the issue of why OS and free software project contributors give their work away into public domain in face of the lack of the intellectual property rights. O’Mahony (2003) emphasizes the role of several ‘legal and normative tactics’ used by projects in protecting the community developed software from proprietary appropriation and in securing their collective identity and reputation. These tactics shed light on how communities govern their work and they are seven. First, adopting software licenses with distribution terms that restrict proprietary appropriation; second, encouraging compliance with licensing terms through normative and legal sanctions; third, incorporating to hold assets and protect individual contributors from liability; fourth, transferring individual property rights to collectively managed non-profit

corporations; fifth, trademarking the brands and logos designed to represent their work; sixth, assigning trademarks to a foundation; and seventh, actively protecting the project's brand (O'Mahony, 2003, p. 1183).

Frank and Jungwirth (2003) distinguish between approaches of 'rent-seekers' and 'donators'. They highlight the two groups of volunteers. The former approach emphasizes the fact that although no wages are paid to OSS contributors, other payoffs may turn their effort into a profitable investment; while the latter approach concerns the many contributors to OSS projects who do not expect to ever receive any individual rewards. They argue that the basic institutional innovation in OS has been the "crafting of a governance structure" which enables rent-seeking without crowding out donations (Frank & Jungwirth, 2003, p. 401). The authors show how "distinct institutional mechanisms" (such as Copyleft and or GNU General Public License) are used by OSS governance structure to reconcile the interests of rent-seekers and donators.

These discussions lead us to appreciate the significant role of governance in the context of OSS development process as a form of a goal-directed network. Therefore, the network of OSS participants, to be effective, needs to be 'governed' through formal and informal coordination mechanisms. Coordination, here, refers to managing a system of exchanges (Levine & White, 1961) which is central to governing and aligning the heterogeneity associated with types of participants and their incentives for participation besides the overall goal of the network. Thus, it is safe to argue that the inclusion of non-software enterprise users into the OSS technological collaborations may introduce new challenges to the governance of the projects and therefore the whole network effectiveness. Consequently, leadership of OSSTC may take a different shape by demonstrating certain OSS-specific capabilities and/or characteristics in order to avoid breakdown of the collaborations and technological systems.

### 3.5.5 Organization of OSS projects

In addition to the significant role of governance in turning OSS projects and networks into a an effective and successful collective effort, one should be wary of the “organization” of OSS innovation process which has been the subject of research during the past decade (e.g., West & O’Mahony, 2005; Dempsey et al., 2002; Scacchi, 2002; Gallivan, 2001).

At the heart of the issue of organization of OSS projects is the “pronounced difference in ‘roles’ taken by contributors” (von Krogh & von Hippel, 2006, p. 979). Nonneke and Preece (2000) highlight the issue of “lurkers” (those who play a passive role) within online communities. Lurkers can account for as high as 90% of online groups despite their role being unclear. Nonneke and Preece (2000, p. 79) reveal that lurking is *not necessarily a negative phenomenon* as it involves a silent participation different from being a free rider. In fact, “a case can be made for lurking being normal and public posting being abnormal. After all, if everyone were posting, who would be reading”. They also show that the actual number of lurkers is much less than what had been reported, widely ranging from zero to 99%.

West and O’Mahoney (2005) distinguish between two types of OS projects: community-founded projects vs. private organization sponsored projects. The former is traditional model of OSS development that is initiated by one or more individuals independent of their employment context<sup>44</sup>. The latter is an alternative OS project model which is called a sponsored or spinout model (West & Gallagher, 2006). Sponsored model concerns the situation where a sponsor of an internally developed software project releases its code to the public under an OSS license, inviting the external community to join the project. The authors compare the two models in order to find out *how sponsored OS projects can best utilize their resources while still managing to grow a diverse external community of contributors*. The authors conjecture that firms that operate with a completely open development process, laissez-

---

<sup>44</sup> Examples include Linus Torvalds’ starting the Linux operating system; Miguel D’Icaza’s initiating the GNOME desktop environment project; and the eight developers who adapted the NCSA web server to become Apache.

faire governance or abdicate community leadership may unintentionally waste resources, hurt their ability to appropriate returns or advance their competition. Further, West and O'Mahoney (2005) show that creating a spinout OS project shares many of the same issues as a community managed one, such as building a collaboration infrastructure, designing governance mechanisms and making key decisions about licensing and other external relationships.

In a follow-up study, West and O'Mahoney (2008) try to investigate how sponsors design OSS communities in the hopes of attracting external participation, and how this differ from the design of autonomous-based communities? Through unboxing the 'design decisions' that sponsors have made when creating a community, authors identify three dimensions that affected participation: 1) the organization of production; 2) governance; and 3) intellectual property. In doing so, West and O'Mahoney (2008) show that the participation architecture of a technical community is determined not only by its technical architecture, but also by community design decisions made by the community's leaders. The aspects of community design that are identified are also critical to attracting and enabling participants. The reason is that they shape the landscape of opportunities extended. Their study, as well as others mentioned above, reveals a 'conflict' or 'tension' that is created when a firm sponsors an OS community project, as there are conflicting goals. On the one hand, firms favor holding control over technologies fundamental to their business success. On the other hand, providing the opportunity structure for others to participate has been a prerequisite for gaining the benefits from developing an external community. Therefore, while designing participation architecture, firms need to take a middle-road position and mediate between surrendering control and offering opportunities for outside participation that could lead to community contributions and growth.

Discussions on organization of OSS projects and OSS development process emphasize how firms and communities are concerned with external participation and sustaining their contributions. Therefore, organization of people is tightly interwoven with success and viability of the whole OSS RDIP. The problem is further exacerbated when we consider augmenting the heterogeneity level of participants by inclusion of or reserving



a role for non-software enterprises from different downstream industries. The implications of including other participants as *technological collaborators* may further impose ‘unexpected organization challenges’ which have not so far identified and addressed in the extant literature.

### 3.6 Conclusion

In this chapter, I have reviewed the notable works on the main research areas of OSS; namely, motivations of firms and individuals; concept of success in the context of OSS; as well as governance, organization and innovation process of OSS collaborative RDIP. I have also discussed the five inherent particularities of OSS technology which collectively build the case for developing a theory that takes into account these attributes. Indeed, we need to be cautious how these attributes unfold in OSS collaborative RDIP and further influence the nature of collaborations, organization of innovation, and governance of projects. This chapter pictures OSS mainly as a public good (non-rival and non-exclusive) which does not suffer from the tragedy of commons. Nonetheless, having some qualities of private goods (ownership and control offered by certain licenses, copyrights, and subscription fees) in case of commercial and enterprise version of OSS can disturb the balance between collective value creation and private value capture.

Although openness of the source code is a shared quality between free software and OSS, the same attribute distinguishes OSS sharply from proprietary software. The past research clearly shows challenges and differences associated with governance and organization of commercial and strategic version of OSS. Particularly, as OSS, in general, is continuously faced with commercialization and productization challenge, we need to further study its development process to make sure all significant stakeholders are included in its RDIP. Such *integrative or holistic approach* to including a broader range of participants such as non-software enterprise users (rarely studied in the extant literature) from downstream industries ensures having a software creation process that



responds to real and immediate market needs. However, this also means that we are adding to challenges of OSS organization and governance. Augmented heterogeneity level in the software production level can translate into more diversity in motivations to engage, various expectations of capture value, more friction and conflicts, and a more uncertain project success. These issues, therefore, call for a more inductive research approach in this field to gain a fresh perspective and provide a finer grained understanding of these issues.

## CHAPTER IV

### RESEARCH METHODOLOGY

#### 4.1 Introduction

This chapter presents the methodological approach adopted to realize this research project. It includes five sections. This introductory note encompasses how this chapter is organized. Section 4.2 explicates the two main methodological approaches (quantitative and qualitative) and explains the rationale behind choosing the qualitative methodology to conduct this research. Further, it explains how three variations of qualitative methodology (case study research, participant-observation, and grounded theory approach) fit well together in the course of the present research. Next, section 4.3 presents the essence of research design by highlighting case and unit of analysis, sampling methods, data collection as well as methods of data analysis. Later, section 4.4 discusses issues like validity, reliability, research ethics and generalizability which are related to evaluating qualitative research. Finally, section 4.5 closes this chapter by highlighting its methodological contribution.

#### 4.2 Methodological approach

Research methodology is a strategy to make an enquiry and it implies underlying philosophical assumptions, research design, and data collection (Myers, 2013). Methodology of research includes the “general strategy” of research (Vogt, 2007, p. 5). It also emphasizes on choosing the right method by comparing advantages and disadvantages of different methods (Vogt, 2007). Research method, on the other hand, is more specific (i.e., it concerns “how” the research is conducted); and, it refers to the “means of accomplishing the ends of using knowledge to answer questions, solve problems, and create knowledge” (Vogt, 2007, p. 5).

Research methodology is typically classified into two major categories: quantitative and qualitative. However, during the past decade, the interest in conducting 'mixed methods research' or 'mixed methodology' has been increasing (Bryman, 2006, p. 97) so much so that Bryman (2006) hold that "we end up with three distinct approaches to research: quantitative; qualitative; and what is variously called multi-methods (Brannen, 1992), multi-strategy (Bryman, 2004), mixed methods (Creswell, 2003; Tashakkori & Teddlie, 2003), or mixed methodology (Tashakkori & Teddlie, 1998) research". Further, Johnson, Onwuegbuzie, and Turner (2007, p. 113) position the mixed methodology somewhere "between the extremes Plato (quantitative research) and the Sophists (qualitative research), with mixed research attempting to respect fully the wisdom of both of these viewpoints while also seeking a workable middle solution for many (research) problems of interest". In this section, I do not deploy mixed methodology approach, as it does not respond to the needs of this study. I shall therefore limit the discussion to the classical qualitative vs. quantitative approaches.

While quantitative research methods originate in the world of natural sciences to investigate natural phenomena, qualitative research methods have long belonged to the realm of social sciences to enable the social scientists to study and understand socio-cultural phenomena. Strauss and Corbin (1990, p. 17) define qualitative research as:

"any kind of research that produces findings not arrived at by means of statistical procedures or other means of quantification. It can refer to research about person's lives, stories, *behavior* [emphasis added], but also about *organizational functioning*, social movements, or *interactional relationships* [emphasis added]. Some data may be quantified as with census data but the analysis itself is a qualitative one".

According to Strauss and Corbin (1990, p. 18), qualitative research is one that involves "nonmathematical analytic procedure that results in findings derived from data gathered by a variety of means. These include observations and interviews, ...".

In further characterizing the qualitative research methodology, Yin (2011, p. 7-8) introduces five important ‘features’ of qualitative research. First, it involves studying the meaning of people’s lives, under real-world conditions; 2) it represents the views and perspectives of the people; 3) it covers the contextual conditions within which people live; 4) it contributes insights into existing or emerging concepts that may help to *explain* human social behavior; and finally 5) it strives to use *multiple sources of evidence* rather than relying on a single source alone.

Tracy (2013, p. 2) introduces the three ‘core concepts’ of qualitative approach as: 1) *Self-reflexivity*; 2) *context*; and 3) *thick description*. The first concept refers to “the careful consideration of the ways in which researchers’ past experiences, points of view, and roles impact these same researchers’ interactions with, and interpretations of, the research scene”. This core element highlights the role of the researcher’s “baggage” or as some call it “wisdom” in the research process; as she or he brings her or his own worldviews into the research. Therefore, “rather than deny our way of seeing and being in the world, qualitative researchers acknowledge, and even celebrate it” (Tracy, 2013, p. 3). Tracy (2013, p. 3) continues by characterizing ‘context’ as the immersing process of qualitative researcher in the research scene where he or she tries to make sense of it. Finally, she views ‘thick description’– which is related to the context– based on the premise that “meaning cannot be divorced from this thick contextual description” (Tracy, 2013, p. 3). For example, a person’s winking could mean a number of things ranging from flirting to an uncontrollable facial twitch (Geertz, 1973; as cited in Tracy, 2013, p. 3). Thus, understanding relationships, groups, and organizations are among main foci of qualitative research. Indeed, qualitative research is capable of providing important insight into interpersonal relationships through interviews and participant observation (Tracy, 2013).

Tracy (2013, p. 25-28) further highlights four ‘main characteristics’ of qualitative research: 1) *Gestalt*; 2) *Bricolage*; 3) *the funnel metaphor*; and 4) *sensitizing concepts*.

First, the 'gestalt' concept "captures people's tendency to piece together various parts into an integrated system or culture. The meaning of these systems comes through their interdependence and integration: the perceived whole is more than a sum of its parts" (Tracy, 2013, p. 26). Therefore, the holistic approach of qualitative researcher makes the phenomenon an understandable one. Second, within qualitative approach, the researcher is painted as "bricoleur"; meaning that qualitative researchers "are like quilters, borrowing and interweaving viewpoints and multiple perspectives.... in order to construct a meaningful, aesthetically pleasing, and useful research synthesis" (Ibid., p. 26). This activity is captured through the concept of "bricolage"; i.e., "a pieced together set of representations that is fitted to the specifics of a complex situations" (Denzin & Lincoln, 2005, p. 4). Third, the funnel metaphor refers to the nature of qualitative research that it usually starts with a broad and wide-open question such as "what is going on here?" and then it narrows down in focus, as the researchers gather more data and "slowly but surely circle through the funnel" (Ibid., p. 27). Therefore, it is important for qualitative researchers to be comfortable with a certain amount of lack of control and having some tolerance for ambiguity. Fourth and last, 'sensitizing concepts' refers to "theories or interpretive devices that serve as jumping-off points or lenses for qualitative study" (Charmaz, 2003; Glaser & Strauss, 1967; as cited in Tracy, 2013, p. 28). Therefore, researchers' past experiences and literature review efforts, can equip them with some "background ideas" which "offer frameworks through which researchers see, organize, and experience the research problem" (Ibid., p. 28).

Furthermore, there are different types of qualitative research, "although no formal typology or inventory exists" (Yin, 2011, p. 17). The author provides ten variations of commonly accepted forms of qualitative research. He further suggests that these variations "do not group into any orderly categories" (Ibid., p. 16). Therefore, as there is an overlap between variations one can conduct a case study research based on participant-observation (Yin, 2011).

These ten types or variations of qualitative research are: 1) *Action research*; 2) *case study*; 3) *ethnography*; 4) *ethnomethodology*; 5) *feminist research*; 6) *grounded theory*; 7) *life history*; 8) *narrative inquiry*; 9) *participant-observer study*; and finally 10) *phenomenological study* (Yin, 2011, p. 17). Yin (2011, p. 16) further emphasizes that “you need to be sensitive to these variations, but you do not need to choose among them if you do not wish to”. However, irrespective of their differences, virtually all qualitative research appears to follow, if not all, of the five features of qualitative research described earlier (Yin, 2011) in this section. Another way to better understand qualitative and quantitative research methodologies is by way of comparison. There are key differences between the two methodologies.

One major distinction between the two methodologies concerns their methods of data collection, data type, analysis and their presentation. Tracy (2013, p. 28) mentions that qualitative method is an “umbrella term” that includes interviews, participant observation, and document analysis. Howard and Prividera (2008) include open-ended qualitative surveys. Tracy (2013, p. 24) provides an acute observation concerning the two. She mentions, “quantitative research transforms data... into numbers.” In fact, quantitative methodologies resort to “measurement and statistics to develop mathematical models and predictions” (Ibid., p. 24). She further goes on and explicates her point by highlighting three key differences. For example, she mentions that quantitative researchers may aggregate data collected through a survey method to measure how often respondents engage in a certain activity. Therefore, “quantitative research is usually driven by questions of scale of the type ‘how much?’ and ‘how often?’” (Ibid., p. 24). Such methodology lacks in thick descriptions of the scene, a quality that characterizes the qualitative research. Similarly, based on Denzin and Lincoln (2000, p. 3), “while quantitative research presents statistical results represented by numerical or statistical data, qualitative research presents data as descriptive narration with words and attempts to understand phenomena in “natural settings”. This means that qualitative researchers study things in their natural settings, attempting to make sense of, or to interpret, phenomena in terms of the meanings people bring to them.” Next, she raises the role each methodology assigns to researcher establishes a

key distinction between the two methodologies. In quantitative research the “research instrument and the researcher controlling the instrument are two separate and distinctly different entities” (Ibid., p. 24); while in qualitative research “the researcher *is* the instrument” (Ibid., p. 25). Finally, the “representation of the methodology, findings, and discussions” of both methodologies is different. In articles following a quantitative methodology, there is tendency to separate the description of research instrument from a report on findings; while in qualitative reports “the description of the research methods often flows into the stories, observations, and interactions collected” (Ibid., p. 25).

Other scholars have highlighted different points of distinction. For example, Stake (1995, p. 37) highlights the “purpose of research” as a point of departure in that researcher’s intention to explain and understand a phenomenon forms the main goal of qualitative research; while quantitative research is mainly used to establish relationships between measured variables. Similarly, Strauss and Corbin (1990, p. 19) clearly mention that one employs qualitative research “to uncover and understand what lies behind any phenomenon about which little is yet known. Such methodology can be used to gain novel and fresh slants on things about which quite a bit is already known. Also qualitative methods can give the [sic] intricate details of phenomena that are difficult to convey with quantitative methods”.

Lastly, the nature of qualitative research tends to be inductive while quantitative research tends to be deductive (see Lincoln & Guba, 1985). Inductive reasoning is a “bottom-up or little-to-big” approach while deductive reasoning refers to a “top-down or big-to-little” approach (Tracy, 2013, p. 21). According to Tracy (2013, p. 21), in qualitative methods we often speak of “emic understandings of the scene”; i.e., “behavior is described from the actor’s point of view and is context-specific”. Emic or inductive, therefore, refers to “meanings that emerge from the field”. On the other hand, in deductive reasoning we speak of “etic understandings, in which researchers describe behavior in terms of external criteria that are already derived and not specific to a given

culture.....a deductive and Etic research begins with External Theories (presuppositions or criteria) to determine and frame meanings” (Ibid., p. 21).

Overall, qualitative approach to research is a scientific methodology which is most suitable when the researcher aims to gain (fresh) understanding and explain an intricate phenomenon which is context dependent. Its data collection methods and data quality allow the researcher to inquire about different perspectives and angles of the same phenomenon and present different meanings as they co-exist in the same context. Its “analytic or interpretive procedures” can be used to develop theories inductively (Strauss & Corbin, 1990, p. 20) and provide thick descriptions to justify the objectivity of the end results of the research.

#### 4.2.1 Rationale for selecting the qualitative methodology and its three variations

The overarching goal of this research is to *develop a theory that explains the nature of the OSS technological collaborations of the shared R&D and innovation process within OSS context*. In order to better understand the conceptual nuances and be able to make a contribution to the existing literature, I need to build a theory ground-up from the industry-based observations. Such theory building process feeds on individuals’ as well as commercial firms’ perceived incentives and motivations to participate in the OSS technological collaborations, their roles and nature of interactions and relationships, among other factors. Literature has placed an emphasis on adopting an inductive approach to theory building in the realm of sciences, including management and psychology (see, Locke, 2007). In fact, this approach to theory development has proven to be an effective and successful one. For example, ‘goal setting theory’ has been ranked number one in importance out of seventy-three management theories by organizational behavior professors (Miner, 2003; as cited in Locke, 2007, p. 877). Since the goal of the present research is theory development, and not theory testing, I have adopted qualitative research methodology governed by inductive approach to reasoning as the dominant methodological approach.



Within the domain of qualitative approach, I then intend to adopt and integrate three approaches: 1) Case study (Yin, 2009); 2) Grounded theory (Glaser & Strauss, 1967; Strauss & Corbin, 1990); and 3) Participant-observer study (Becker, 1958; DeWalt & DeWalt, 2002).<sup>45</sup>

#### 4.2.2 Case study

*Case study* is one of several ways to conduct research in the field of social science. It “is an empirical inquiry that investigates a contemporary phenomenon in depth and within its real-life context, especially when the boundaries between phenomenon and context are not clearly evident” (Yin, 2009, p. 18). Furthermore, Yin (2009) highlights the usual inseparability of phenomenon and context in real life situations. Case study “copes with the technically distinctive situation in which there will be many more variables of interest than data points, and as one result relies on multiple sources of evidence, with data needing to converge in a triangulating fashion, and as another result benefits from the prior development of theoretical propositions to guide data collection and analysis” (Ibid., p. 18). Therefore, it represents an “all-encompassing method-covering the logic of design, data collection techniques, and specific approaches to data analysis” (Ibid., p. 18).

Yin (2009, p. 8) argues that there are three necessary conditions underlying a method selection: a) *research question type* (Hedrick, Bickman, & Rog, 1993; as cited in Yin, 2009, p. 9); b) “the *extent of control* an investigator has over actual behavioral events”; and c) “the *degree of focus* on contemporary as opposed to historical events.”

Yin (2009, p. 8-10) provides a basic category of questions (i.e., ‘who’, ‘what’, ‘where’, ‘how’, and ‘why’) and further explains how a researcher can match the type of questions with different methods of investigation (i.e., experiment, survey, archival analysis, history, and case study). Based on his arguments, for the research questions posed in the present study, case study is the most appropriate method of investigation.

---

<sup>45</sup> Please refer to Yin (2011, p. 17) to gain a detailed understanding on variations of qualitative research.

For example, the questions which start with “how” and “why” in this study are *explanatory* in nature. These questions, normally, “deal with operational links needing to be traced over time, rather than mere frequencies or incidence” (Ibid., p. 9). Even the questions that start with “what” in this study are not designed as variations of “how many” or “how much” lines of inquiry; they indeed are *exploratory* in nature and they are designed with the goal of developing pertinent propositions and hypotheses. Thus, based on the types and nature of questions, selecting the case study method is an appropriate choice.

Furthermore, as Yin (2009, p. 11) puts it, “the case study is preferred in examining contemporary events, but when the relevant behaviors cannot be manipulated”. In this research, I deal with living OSS developers and investigate the process of OSSD as it is happening. Therefore, my focus is on *contemporary events* and not those considered as “dead past”. In addition, as a researcher, I cannot manipulate the participants’ behaviors. Thus, these two conditions lead me to choose case study as the dominant research approach. Equally important, case study method is a strong method as it deals with variety of evidence so much so that it relies on two important sources of evidence that are direct observation of the events being studied as well as interviews with the people involved in those events (Yin, 2009). These sources of evidence also pave the way to integrate the ‘participant observation’ method with classical case study approach.

#### 4.2.3 Participant observation

*Participant observation* is a method through which researchers generate understanding and knowledge by watching, interacting, asking questions, collecting documents, making audio or video recording, and reflecting after the fact (Lofland & Lofland, 1995). It denotes a participatory approach to research in which method “a researcher takes part in the daily activities, rituals, interactions, and events of a group of people as one of the means of learning the explicit and tacit aspects of their life routines and their culture” (DeWalt & DeWalt, 2002, p. 1).

Participant observation is also known as “*fieldwork*” (see, Tracy, 2013, p. 65; DeWalt & DeWalt, 2002, p. 1). According to Delamont (2004, p. 218) fieldwork refers to “the data collection phase, when the investigators leave their desks and go out ‘into the field.’ The field is metaphorical: it is not a real field, but a setting or a population”. Later, Tracy (2013, p. 65) adds the term “fieldplay” based on her experience. The term implies: “adventure, curiosity, and playfulness” as parts of the participant observation. In fact, “*fieldplayers* are filled with good humor, improvise, and do not take themselves too seriously” (Tracy, 2013, p. 65).

Agar (1996) uses the term participant observation “as a cover term for all of the observation and formal and informal interviewing in which anthropologists engage” (as cited in DeWalt & DeWalt, 2002, p. 2). Participant observation is “the starting point in ethnographic research” (Schensul et al., 1999, p. 91) in which sense it is considered as the foundation for ethnographic research (Schensul et al., 1999).

DeWalt & DeWalt (2002, p. 4) highlight *seven key elements* of participant observation (as used by anthropologists) as follows: 1) living in the context for an extended period of time; 2) learning and using local language and dialect; 3) actively participating in a wide range of daily, routine, and extraordinary activities with people who are full participants in that context; 4) using everyday conversation as an interview technique; 5) informally observing during leisure activities (hanging out); 6) recording observations in field notes (usually organized chronologically); and 7) using both tacit and explicit information in analysis and writing.

Participant observation is viewed as an ‘advantageous’ method of inquiry in conducting research for three main reasons.<sup>46</sup> Firstly, “it enhances the quality of the data obtained

---

<sup>46</sup> However, this is subject to debate mainly due to some historical ‘mistakes’ that have taken place in the field of anthropology. For instance, the methodology (i.e., participant observer) and findings of the highly celebrated work of the renowned American anthropologist: *Margaret Mead* known as “*Coming of Age in Samoa*” – a book based upon the study of primarily adolescent girls on the island of Ta’u in the Samoan Islands – was later severely criticized by Derek Freeman. Freeman’s critique is based on his claim that Mead’s informants have lied to her and therefore her observations are based on incorrect data.

during fieldwork”; secondly, “it enhances the quality of the interpretation of data, whether those data collected through participant observation or by other methods” (DeWalt & DeWalt, 2002, p. 8). Thus, participant observation is both a data collection and an analytical tool. Thirdly, it encourages the researcher to formulate new research questions and hypotheses which are grounded in the on-the-scene observations (DeWalt & DeWalt, 2002, p. 8).

Participant observation is a qualitative method which is also tightly interwoven with case study research and interpretive research paradigm within information systems’ research (see, Klein & Myers, 1999). With the advent of interpretive research as a significant strand in the field of information systems’ research (see, Walsham, 1995b); scholars (Klein & Myers, 1999, p. 67) believe that it can aide IS researchers to better “understand human thought and action in social and organizational contexts” as such paradigm “has the potential to produce deep insights into information systems phenomena”. However, case study can be conducted under different philosophical paradigm; it can be positivist (Yin, 1994), interpretive (Walsham, 1993), or critical (Klein & Myers, 1999). Therefore, to further clarify the scope of case study within interpretative research and more specifically “interpretive field studies” (Walsham, 1993), Klein and Myers (1999, p. 71-79) propose *seven principles*, based on practice of anthropological research as well as underlying philosophy of phenomenology and hermeneutics, to evaluate interpretive field research.

First, “the fundamental principal of the hermeneutic circle” suggests, “all human understanding is achieved by iterating between considering the interdependent meaning of parts and the whole that they form” (Ibid., p. 72). Second, the principle of contextualization “requires critical reflection of the social and historical background of

---

Source: based on: Wikipedia. *Coming of Age in Samoa*. Retrieved from [https://en.wikipedia.org/wiki/Coming\\_of\\_Age\\_in\\_Samoa#Critique\\_of\\_Mead.27s\\_methodology\\_and\\_conclusions](https://en.wikipedia.org/wiki/Coming_of_Age_in_Samoa#Critique_of_Mead.27s_methodology_and_conclusions). Note: These anecdotal examples shed light on weaknesses of participant observer’s method, albeit they are also based on another participant observer’s view (e.g., Derek Freeman). However, the *triangulation technique* can reduce the margins of error to a great extent. More specifically, because OSS developers are not like the youth of Samoan Islands concerning their *free will* and *freedom of choice*, and since there is a *large number* of them, dispersed around the globe, we can reduce the possibility of lying through using multiple data sources.

the research setting, so that the intended audience can see how the current situation under investigation emerged” (Ibid., p. 72). Third, the principle of interaction between the researchers and the subjects “requires critical reflection on how the research materials (or ‘data’) were socially constructed through the interaction between the researchers and participants” (Ibid., p. 72). Fourth, the principle of abstraction and generalization “requires relating the idiographic details revealed by the data interpretation through the application of principles one and two to theoretical, general concepts that describe the nature of human understanding and social action” (Ibid., p. 72). Fifth, the principle of dialogical reasoning “requires sensitivity to possible contradictions between the theoretical preconceptions guiding the research design and actual findings (‘the story which the data tell’) with subsequent cycles of revision” (Ibid., p. 72). Sixth, the principle of multiple interpretations “requires sensitivity to possible differences in interpretation among the participants as are typically expressed in multiple narratives or stories of the same sequence of events under study. Similar to multiple witness accounts even if all tell it as they saw it” (Ibid., p. 72). Seventh, the principle of suspicion “requires sensitivity to possible ‘biases’ and systematic ‘distortions’ in the narratives collected from the participants” (Ibid., p. 72).

These seven principles clearly show the inter-connection between the three issues: case study, participant observation as fieldwork, and interpretivisms as school of thought. Case study places emphasis on studying the phenomenon as it lives in a real-life context through collecting data through multiple sources of evidence. It is an initiative to get intimate with what a researcher studies. Participant observation allows for this intimacy to shape as the researcher becomes engaged in the field by becoming part of the research instrument with heightened interest and awareness. Indubitably, principle of ‘interaction between the researchers and subjects’ captures so well the existing connections, since it maintains that “in social science, the ‘data’ are not just sitting there waiting to be gathered, like rocks on the seashore. Rather, interpretivism suggests that facts are produced as part and parcel of the social interaction of researchers with participants (Klein & Myers, 1999, p. 74)”. After all, as Becker and Geer (1957, p. 28) put it:

“The *most complete form of the sociological datum*, after all, is the form in which the *participant observer gathers it*: An observation of some social event, the events which precede and follow it, and explanations of its meaning by participants and spectators, before, during, and after its occurrence. Such a datum gives us more information about the event under study than data gathered by any other sociological method.”

Lastly, Becker and Geer (1957, p. 28-29) distinguish between participant observation and interviewing as two methods within the camp of qualitative methodology with regards to three main issues: “learning the native language, or the problem of the degree to which the interviewer really understands what is said to him; matters interviewees are unable or unwilling to talk about; and getting information on matters people see through distorting lenses”.

The first issue refers to complexities associated with language of the social group being studied. It is assumed that the language is a socio-cultural phenomenon that has its own subtleties like words with different nuances of meanings, denotations and connotations. These differences rooted in contextual matters form sources of confusion, error in dialogue and improper understanding on the side of interviewer and interviewees: “The interview provides little opportunity of rectifying errors of this kind where they go unrecognized” (Becker & Geer, 1957, p. 29). Therefore, it is believed that participant observer can overcome these issues mainly because:

“participant observation provides a situation in which the meanings of words can be learned with great precision through study of their use in context, exploration through continuous interviewing of their implications and nuances, and the use of them oneself under the scrutiny of capable speakers of the language” (Becker & Geer, 1957, p. 29).

Second, according to Becker and Geer (1957, p. 30), “frequently, people do not tell an interviewer all the things he might want to know”; and this can be due to a variety of reasons ranging simply from “resistance” to issues like “unfamiliarity”: i.e., some events may be so unfamiliar or vague to interviewees that they may “find it difficult to put into words their vague feelings about what has happened”. Such issues can be

obviated through participant observation strategy as the participant-observer develops his knowledge through observation and interaction with the environment and becomes able to 'make sense' of "vague statements as clues to an objective situation" and or "can connect his knowledge with these half communications".

Third, usually individuals develop their own perceptions about a phenomenon so much so that there are differences in perception, and these will naturally affect what they report in an interview (Becker & Geer, 1957). The strength of participant observation over interview method lies in its ability of the participant-observer "to check description against fact and, noting discrepancies, become aware of systematic distortions made by the person under study; such distortions are less likely to be discovered by interviewing alone" (Becker & Geer, 1957, p. 31).

Overall, participant observation is a method the deals with the core issue of "errors of inference"; i.e., "errors which arise from the necessity of making assumptions about the relation of interview statements to actual events which may or may not be true" (Becker & Geer, 1957, p. 31).

Within the context of OSSD, I am looking at the relationships among the participants (R&D engineers, clients, and community members) in order to understand how they influence the development of strategic OSS. As OSS is a socio-economic technological phenomenon, the incentives, and motivations of participants can be influenced by pecuniary and non-pecuniary elements and the process of technology development is a complex one. Therefore, approaching participants (potential interviewees) without building a rapport may result in collecting data that might be merely a demonstration of the surface of the issue at hand. Further, as discussed above, due to shortcomings of interview method, the quality and accuracy of data will not be at a sufficient level to lead to high level of reliability and validity of results. Therefore, adopting the participant-observer strategy will facilitate my integration into the context and fabric of the field and be able to study the OSSD process from an insider's viewpoint.



#### 4.2.4 Grounded theory approach

Grounded theory is originally developed by the sociologists: Barney Glaser and Anselm Strauss (1967), and it was later extended by Strauss and Corbin (1990) as well as Charmaz (2006). Grounded theory approach “is a qualitative research **method** that uses a **systematic** set of **procedures** to develop an inductively derived grounded **theory** about a **phenomenon**” (Strauss & Corbin, 1990, p. 24, emphasis kept as in the original format). The research findings, therefore, constitute a theoretical formulation of the reality under investigation, rather than consisting of a set of numbers, or group of loosely related themes (Ibid., p. 24). The main goal of grounded theory approach (GTA) is “to build theory that is faithful to and illuminates the area under study” (Ibid., p. 24). Grounded theory building approach leads to creation of a kind of theory that is inductively derived from the study of the phenomenon it represents. This means that the theory is “discovered, developed, and provisionally verified through systematic data collection and analysis of data pertaining to that phenomenon” (Ibid., p. 23). Tracy (2013) refers to the systematic inductive analysis of data from ground up in form of a bottom-up data analysis.

Scholars who aim to develop a theory rather than testing an existing one believe that forming theoretically informed interpretations is the most powerful way to bring reality to light (Blumer, 1969; Diesing, 1971; Glaser, 1978). Following the assumption that reality cannot be actually known by man but it is always interpreted, “building theory, by its very nature, implies interpreting data, for the data must be conceptualized and the concepts related to form a theoretical rendition of reality” (Strauss & Corbin, 1990, p. 22). Such theory is useful not only with regards to explaining the reality we seek to understand, but also with regards to providing a framework for action. Furthermore, Strauss and Corbin (1990) state that, to theory builders, theories represent the most systematic way of building, synthesizing, and integrating scientific knowledge (see p. 22).



In addition, Strauss and Corbin (1990, p. 23) touch upon the four central criteria for evaluating the applicability of grounded theory; these are: *fit*, *understanding*, *generality*, and *control* (see Glaser & Strauss, 1967, pp. 237-250).

First, “if the theory is faithful to the everyday reality of the substantive area and carefully induced from diverse data, it should fit the substantive area”. Next, since such theory represents the reality, it should also be comprehensive and make sense both to the persons who were studied and to those practicing in that area. In addition, if the researcher uses “comprehensive data”, and the “interpretations are conceptual and broad”, then the developed theory tends to be abstract enough to include sufficient variation to make it applicable to a variety of contexts related to that particular phenomenon. Lastly, the developed theory should provide control with regard to action toward the phenomenon. Control is important because the outcome hypotheses, which state the nature of relationship among concepts, should be systematically derived from actual data that are only related to the phenomenon under investigation. These hypotheses will guide further action concerning the phenomenon. Besides, the conditions to which the theory applies need to be clearly spelled out. (Strauss & Corbin, 1990, p. 23)

Within GTA, the researcher follows an emic or inductive approach to data collection and analysis. This means that she or he, instead of approaching the data with imposing prior theories on the data to be analyzed, begins the research process with data collection stage and engages in open line-by-line analysis so as to create themes and link them together in a larger story (Tracy, 2013, p. 30). Following this approach, the researcher does not impose a framework of his own on the study participants, therefore the chances to distort their ideas are reduced (see Harris, 1976). Therefore, the research process starts from individual cases and from incidents in the data, and these develop progressively into more abstract categories and theories (Tracy, 2013, p. 184). Taking such approach to theory development requires the researcher to postpone the literature review until after the data are collected; as the theory is data driven and not from research questions or existing literature (Tracy, 2013, p. 184). Furthermore, within

GTA, data analysis is not delayed until all data are collected. Rather, data collection and analysis take place concurrently; meaning that the researcher studies the data as they are collected (Charmaz, 2006, p. 80).

Overall, as the purpose of the present research is to further develop our theoretical understanding of the nuances of OSS technological collaborations. To this end, I have selected the GTA as a variation of the broader qualitative approach guiding this piece of research. Indeed, “the major difference between this methodology [GTA] and other approaches to qualitative research is its merits upon theory development” (Strauss & Corbin, 1994, p. 274). Furthermore, the choice of participant observation method fits this methodology very well as it enables the researcher to collect and evaluate data based on everyday reality and change direction of data collection more appropriately, conveniently and swiftly so as to follow the principle of GTA which is mainly data driven as opposed to theory driven. Also, by collecting diverse data from different industry participants such as OSS engineers, OSS firms, developers of free and OS communities and clients of OSS firms, I can make sure the diversity or comprehensiveness of data based on which the theory is being developed.

Finally, I need to address some concerns about the adoption of GTA in this research. Indeed this dissertation did start with two chapters of literature review nature; while principally, according to GTA, data collection stage precedes theory analysis. My justifications to rely, partly, on GTA rest in the following argument. First, I have written Chapter 2, which presents a comprehensive literature review of OI, in order to situate the phenomenon of interest through highlighting the weaknesses, strengths and theoretical gaps of OIP. This chapter provides the much-needed context for me to more clearly define and formulate the research problem about OSSTC. OSSTC is a phenomenon in search for a clearer, more detailed and more independent theoretical understanding, one that does not necessarily depend on OI framework. Second, Chapter 3, which provides the current understanding and perspectives on OSS, informs us of conceptual cracks and blind spots that need a finer-grained explanation. For instance, it demonstrates that the role of OSS commercial clients in the shared R&D and

innovation process has not been widely studied in the past (at least to the best of my knowledge). It emphasizes consideration of several factors such as the changing landscape of OSSD and its innovation process, the high failure rate of OSS projects, the critical role of corporate consumers in providing the whole-product concept of a market-driven business approach while developing theories about OSS shared RDIP. The literature reviewed in this chapter *does not* tell us a theoretical story about the details of how clients can actually influence OSSD process or how OSSD process benefits downstream commercial industries, if commercial users get actively engaged in shared RDIP. Thus, I use these literature-based insights to be able to cast a wider net and collect more and different data from the OSS industry ecosystem. A metaphor will perhaps clarify this issue further.

It is my understanding (and I do not claim that it can be all right and fine) that when a researcher follows the tradition of GTA to theory development, he resembles a *fisherman* who is about to venture deep into the unpredictable ocean. He is completely blind on how the sea will eventually treat him or how many fish, after all, he will catch. Therefore, to reduce his uncertainty, the night before he sails off, he goes to the local beach cabin for a chat with some fellow fishermen. His evening discussion with a few at the cabin the night before sailing off may lead him to choose a *wider* and *thicker* net to increase his chances of success. However, carrying a wider or thicker net does not necessarily mean that he possesses enough knowledge about how the adventure will unfold, or how many fish will fall into his prey. Thus, the Chapter 3 is more of an evening discussion with other sailors that provides some theoretical tips rather than deep theoretical discussions on the actual research inquiry.

Finally yet importantly, based on the inductive results of Chapter 5, I will actually go and revisit the existing theories in depth in order to figure out whether the inductive theory that I tried to develop will bear novelty vis-à-vis the existing perspectives. By way of comparison, I carve out the nuances of differences that exist between the findings in this research and those of the existing theories. By doing so, I can provide a finer-grained explanation as the theoretical contribution of this research. GTA, in this

sense, helps me to provide a detailed theoretical perspective that did not exist in its current shape in the existing literature.

#### 4.3 Research design

Yin (2011, p. 75) holds that “every research study has a design, whether implicit or explicit”. Research designs function as “*logical* blueprints” or “plans”, different from “logistics plans” (Yin, 2011, p. 75; see also, Yin, 2009, p. 26). This logic encompasses the “links among the research questions, the data to be collected, and the strategies for analyzing the data” (Yin, 2011, p. 76). Therefore, the aim of the logical blueprint is to ensure that the findings will address the intended research questions (Yin, 2011). However, logistics are concerned with the process of research management such as scheduling and coordinating research work (Yin, 2011).

Overall, “colloquially”, as expressed by Yin (2009, p. 26), a research design refers to “*a logical plan for getting from here to there*, where *here* may be defined as the initial set of questions to be answered, and *there* is some set of conclusions (answers) about these questions”. Put simply, it “guides the researcher in the process of collecting, analyzing, and interpreting observations” (Nachmias & Nachmias, 1992, pp. 77-78).

As this research follows a qualitative methodology, I focus on perspectives of qualitative research design. Qualitative research follows a rather unorthodox design. Yin (2011, p. 77) mentions that “not all qualitative studies start by having a research design”; in fact, “how much design work is done beforehand is a matter of choice” (Ibid., p. 76). Even within the camp of qualitative research, sometimes researchers favor the option of putting fieldwork first and then designing the research questions (see Yin, 2011, Chapter 4).

In the present research, I started the research process by conducting five pilot interviews (over the phone and Skype®) with industry practitioners such as R&D managers of OSS small and medium sized enterprises (SMEs) located in Montreal,

Waterloo and Ottawa, as well as policy makers stationed in Quebec City. My main objective was to better understand how to approach the OSS industry and OSSD process, as well as to be able to design several preliminary research questions. At this pilot stage, I adopted no elaborate or ostentatious research design. However, having completed the pilot phase, I then set forth on preparing the actual research proposal and decided to craft a rather comprehensive research design in order not to get lost amidst conducting the fieldwork; thereby, I revisited the design as the research was progressing. Yin (2011, p. 77) states that “design process is a recursive one”. This is mainly because “qualitative research permits and in some ways encourages multiple midstream adjustments throughout the study process...”. As a case in point, I started the case study with “unstructured interviews” to be able to pose open-ended questions and let the interviewees feel at home and speak their minds freely; but then as the research progressed and goals became clearer then I also used “structured” questions. Structured questions were addressed to certain experts in the field in order to elicit from them answers that were more precise.

#### 4.3.1 Defining the case and the unit of analysis

Defining the unit of analysis is a fundamental yet a challenging stage in conducting the qualitative case study research. It is “related to the fundamental problem of defining what the ‘case’ is- a problem that has plagued many investigators at the outset of case studies (see, e.g., Ragin & Becker, 1992; as cited in Yin, 2009, p. 29). Like a case, unit of analysis can be an individual, an event, or even as broad as a country’s economy (Yin, 2009, p. 29-30). As challenging as it may sound, determining the unit of analysis in the research is related to research questions and research goals; and its selection affects the research design (Yin, 2009). Further, “the units need to be an appropriate reflection of the main topic of study” (Yin, 2011, p. 83).

Yin (2011, pp. 82-87) discusses the subject of unit of analysis under the term “data collection unit” as it represents a “nontechnical reference” which “is usually the unit of analysis, [yet] there are complicated situations when it is not”. All in all, majority of

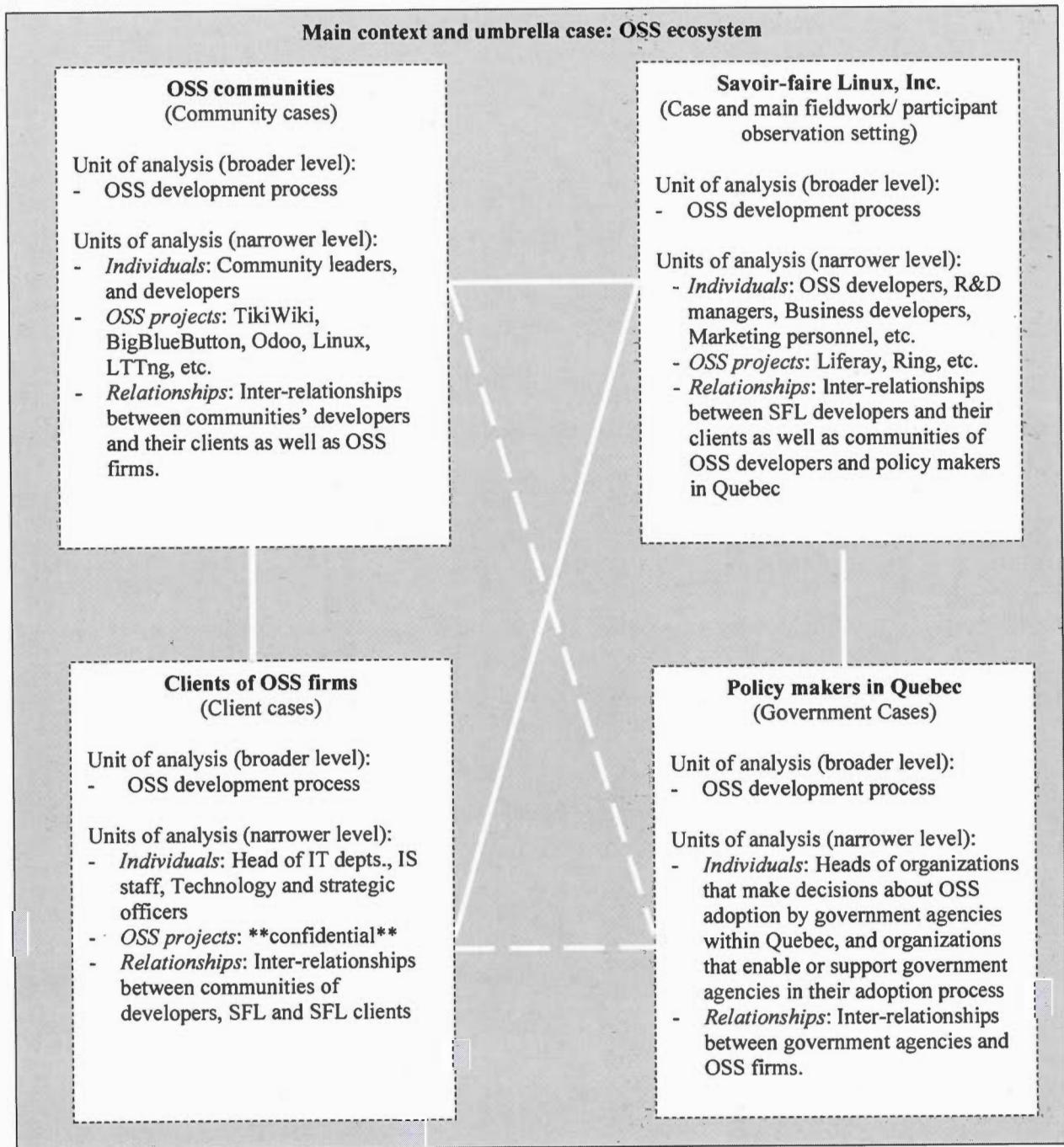
qualitative studies have more than one level of data collection unit and this involves a “broader level” like the case of field setting and a “narrower level” like the participants in the setting (Yin, 2011, p. 82-83). Broader levels include, for example, a geographical region, a country, a factory, a pet store; while the narrower levels can refer to policies, actions, individuals, project teams, communities and schools, and so on (for more examples, see Yin, 2011, pp. 84-85). According to Bizzi and Langley (2012, p. 228), while commenting on how to study processes within the context of networks, many studies rely on multiple levels and units of analysis which can be both spatial and temporal.

In this research, the *broader level* of unit of analysis involves the “process of OSS development” in ICT sector. This process further involves different groups and individuals, projects and communities of software developers, OSS firms, IT consultants, and the inter-relationships that exist among them. Therefore, these individuals and communities with their inter-relationships embedded in their everyday actions form the *narrower level* of unit of analysis. Since, the overall process of OSS development and its participants are part of the OSS ecosystem, therefore, the case under investigation is “OSS ecosystem” which includes OSS firms, OSS community of software developers, commercial clients and users, as well as policies which influence the software development process as well as participants’ behaviors. However, it is not feasible to conduct an in-depth analysis of all elements within this ecosystem. Thus, I have decided to choose a very fertile, active, involved, and conducive OSS firm with a unique business model as the main case setting and branch out into other cases such as OSS communities, policy makers, clients of OSS firms, leaders of OSS communities, R&D managers of other OSS firms in Canada and abroad (France, Belgium, etc.). The pilot studies that I conducted has served the case selection process well. I have finally chosen Savoir-faire Linux (SFL) Inc. as the firm to be embedded in as ‘*industry research intern*’ in order to study the OSS ecosystem up close, to study SFL in-depth, and to capitalize on the relationships between SFL and other entities in the ecosystem. The ties of SFL with the other participants of the ecosystem are of paramount



importance as they can be my gateway to other cases and expand my case study research. Figure 4.1 below illustrates the case and units of analysis schematically.

Figure 4.1. Type of Case Study Design Including Units of Analysis



As Figure 4.1 shows, all cases are considered as an entity within the OSS ecosystem. However, I acknowledge that each case has its own individual context. For example, a client of OSS firm has its own internal bureaucracies, logic, fears and reservations, incentives, and capabilities, to mention a few, to either opt for OSS, proprietary or a mix of the two solutions in order to satisfy their ISs' needs. In addition, when a client adopts an OSS solution and integrates it into their ISs, the client becomes part of the OSS ecosystem (explicitly or implicitly) as the underlying OS product carries the characteristics of generic OSS. Therefore, in this research the client enterprises, communities, and OSS firms are perceived as if they are situated within the context of OSS ecosystem. Government bodies however have their own context.

Since government agencies indirectly influence or are related with OSS communities and private enterprise clients, their relationship is shown through dashed lines; however, SFL's long lasting and effective interactions with government agencies, communities and private client enterprises underline the logic for using solid lines.

#### 4.3.2 Sampling methods

This study adopts two sampling techniques: a) *purposive* (or purposeful) sampling technique; and b) *snowball* sampling technique.

The goal of purposive sampling technique, according to Yin (2011, p. 88), is to "yield the most relevant and plentiful data"; and according to Kuzel (1992, p. 37), it is "to obtain the broadest range of information and perspectives on the subject of study". This method is especially useful because the researcher can choose participants that may offer contradictory evidence or views that are helpful in evaluating the rival explanations (Kuzel, 1992, p. 88). Yin (2011) further recommends that researchers deliberately interview those whom he or she suspects might hold different views related to the topic under investigation. However, one should not confuse purposive sampling with *convenience* sampling as the latter involves "selecting data collection units simply



because of their ready availability” which is not normally a preferred technique (Yin, 2011, p. 88).

The second technique, *snowballing*, refers to “selecting new data collection units [e.g., participants] as an offshoot of existing ones” (Yin, 2011, p. 89). This method, is acceptable only if it is done purposefully and not out of convenience (Yin, 2011).

#### 4.3.3 Data collection methods: sources of evidence

This research uses three data collection techniques to gather evidence on OSS technological collaborations: interviews, participant observation, and documentation.

#### 4.3.4 Interview types

Interviews are an essential source of case study evidence because most case studies are about human affairs or behavioral events (Yin, 2009, 108). Yin (2009, pp. 107-108) divides case study research interviews mainly into “in-depth” interviews; “focused” interviews; and survey like interviews which entail more structured questions. Yin (2011, pp. 132-135) classifies interviews into two categories: *structured* interviews and *qualitative* interviews. Tracy (2013, pp. 138-141) distinguishes among interview structure and type; where she breaks down structure into *structured* and *unstructured* interviews; and types into six including *informant* interviews, *respondent* interviews, *ethnographic* interviews, *narrative* interviews, *life-story* or *biographic* interviews, and *discursive* interviews.

In this study, I conduct the two main types of interviews which are common in case study research; these are “*in-depth interviews*” and “*focused interviews*” (Yin, 2009, p. 107).

I use *in-depth interviews* in order to find out about facts and opinions about OSSD process; the nature of relationships within the technological collaborations; and whatever topics that interviewees find important and critical from their own perspectives. Some interviews extended over a few sessions because they could not be completed in a single setting due to several reasons. For example, sometimes the scope of discussion got too broad and as I did not want to stop interviewees from telling me the whole story, we had to continue on different occasions. Other times, they had to attend to their daily project matters and come back to me on another occasion. I remember one time after about an hour or so of an interview, the interviewee was interrupted by a client call and we had to resume our discussion after a break. In this situation, as I was embedded in the company, scheduling another interview session was not a difficult task to do, though time consuming. There were occasions where my interviewees turned into “*key informants*” rather than being a mere respondent. These informants were those with extensive experience (more than 10 years) in the OSS industry and they could provide me with deep insights about the subject matter. Yin (2009) views the *key informants* as often critical to the success of a case study yet he cautions the researchers of pitfall of becoming over dependent on these key informants. In order to avoid this situation, I followed the recommendation made by Yin (2009) and tried to corroborate their opinions through finding other sources of evidence and searching for contrary evidence as carefully as possible.

I also use *focused interviews* (Merton et al., 1990) which involves interviews of rather shorter periods such as one hour (Yin, 2009). These interviews are performed in a “conversational manner” while remaining open-ended (Yin, 2009, p. 107). The difference is that the researcher can follow certain set of carefully worded questions in order to gain a fresh commentary about the topic; they are especially useful for corroboration purpose (Yin, 2009, p. 107). Since I was embedded within the organization for one year, I managed to conduct many of these focused interviews with even “walk-in” experts. For example, one day a project manager from Microsoft office in Toronto walked into the company to attend a meeting. Upon completion of the meeting, I was invited to have lunch with the guest and some other staffers. This was a

unique opportunity to conduct a focused interview during the long lunch break and jot down the specific points on my pocket research notebook. The so-called “pop-up” interviewee was so nice and cooperative that we continued the interview after lunch in the head office and turned the interview into an informal conversation with open-ended questions.

Furthermore, I used both *structured* and *unstructured* interviews. *Structured interviews*, according to Yin (2011, p. 133) are those in which the researcher uses “a formal questionnaire that lists every question to be asked”. He or she also adopts “the role of an interviewer, trying to elicit responses from an interviewee” (Ibid., p. 133). Finally, the interviewer makes an effort to adopt “the same consistent behavior and demeanor when interviewing every participant” (Ibid., p. 133). In these interviews, the interviewer tends to use more closed-ended questions. On the other hand, *unstructured interviews* – which are referred to as “*qualitative interviews*” by Yin (2011, p. 134) – reflect the nature of qualitative methodology best (Yin, 2011). This is mainly because the relationship between interviewer and interviewee is “not strictly scripted”; although the researcher has a “mental framework of study questions”; he or she does not follow from a questionnaire bearing complete list of questions (Yin, 2011, p. 134). Further, the researcher does not adopt “any uniform behavior or demeanor for every interview”; therefore, the interview follows a “conversational mode” which can “lead to a social relationship of sorts, with the quality of the relationship individualized to every participant” (Yin, 2011, p. 134). Lastly, questions used by the interviewer during a qualitative interview are of open-ended nature (Yin, 2011, p. 135).

In the course of this research, I started with unstructured or qualitative interviews for the following reasons. First, I did not want to bring into the interview session my own prejudgments and already established worldviews – in this case, for example, about the process and methodology of OSS development and the nature of technological collaborations. Second, I intended to discover new concepts and build a theory bottom-up, therefore, I wanted the interviewee to more talk about the sides of the story that were unheard, or less often discussed. In order to describe their experiences, I decided

to start my conversation with how they started to get to know OSS world and how they became engaged. As time went by, I then moved towards more details. Finally, I used qualitative interviews to form a global understanding based on the contextualized knowledge so as to identify a clearer path and direction for further interviews.

Later, as the research advanced, I shifted towards adoption of more semi-structured and sometimes structured questions having a set of questions on mind. The main objective of asking the structured questions was to obtain different perspectives on the repeated themes and be able to corroborate the findings, and highlight the discrepancies and contradictory areas. These discrepancies and contradiction served the research well in highlighting newly emerging themes.

Lastly, all interviews have been conducted under the “Interview Guide” (or UQAM’s interview protocol) created for the present research. UQAM’s interview protocol includes sections such as: *Interview and data management methods* (e.g., interview objectives and key concepts, interview types, duration and number of interviews, interview site, informed consent of participants, anonymity, confidentiality, data security management including cryptography and backup filing); *interview procedure* (e.g., preamble, administration of questionnaire and interviews, closure); *sample of interview protocol forms* (preamble checklist, interview questions, and closure checklist); *data transcription procedure* (e.g., voice file arrangement, confidentiality and data management, transcription site); *flowchart of the whole interview process*; *consent forms in English and French*; *Non-disclosure agreement*; and finally, the essay which has been part of the *training of research team on ethics for research with human subjects*. It took me about three months to prepare, revise, and submit the final version of all these documents to the UQAM’s ethics committee. Having evaluated all the documents, UQAM has issued “certificat d’éthique”<sup>47</sup> (i.e., Ethics Certificate) which allows the researcher to embark on the research journey and enter the field.

---

<sup>47</sup> Certificate issued by the *Comité institutionnel d’éthique de la recherche avec des êtres humains* de l’UQAM.

#### 4.3.5 Research participants

In this study, I have purposefully chosen Savoir-faire Linux as the main case because the company:

- (a) is highly active with regards to contributing codes back to community of OSS developers (e.g., the Linux kernel);
- (b) is developing and launching OSS projects in-house and builds community of developers around them (e.g., Ring and SFLvault projects);
- (c) is extensively involved in promoting OSSD methodology and adoption of OSS solutions through enlightening and enabling enterprise clients (e.g., information and training sessions) and actively participating in public seminars and conferences (e.g., Platinum Sponsor de DebConf17; FOSDEM 2017 & 2016, etc.);
- (d) is playing an active role with regards to promoting the adoption of OSS solutions by the government agencies of Quebec (i.e., playing an evangelical role);
- (e) has taken meaningful steps to stabilize the position of OSS solutions vis-à-vis proprietary software solution within the software industry in Quebec.<sup>48</sup>

Next, to prepare the first list of potential interviewees, I have consulted the chief technology officer (VP for technologies) and management team of SFL in order to choose the participants (i.e. other cases) that are information-rich and highly relevant. Later, as I developed more familiarity with the overall context and study subject (as an embedded researcher within real-life events) coupled with my ongoing experience gained through data analysis parallel to data collection, I had the chance to approach respondents and use the contextual clues in order to get referrals to approach other

---

<sup>48</sup> See the following links for Savoir-faire Linux, Inc. vs. Régie des Rentes du Québec:

- a) Savoir-faire Linux inc. c. Régie des rentes du Québec, 2010 QCCS 2375 (CanLII)  
Source: Savoir-faire Linux inc. c. Régie des rentes du Québec, 2010 QCCS 2375 (CanLII).  
CanLII. Retrieved from  
<http://www.canlii.org/fr/qc/qccs/doc/2010/2010qccs2375/2010qccs2375.html>
- b) Francois2. (2008, March 20). Cour supérieure: Savoir-faire Linux contre la RRQ. *Direction Informatique*. Retrieved from  
<http://www.directioninformatique.com/cour-superieure-savoir-faire-linux-contre-la-rrq-4/9191>
- c) Jean-François Ferland, J-F. (2010, June 04). Savoir-Faire Linux obtient une victoire contre la RRQ. *Direction Informatique*. Retrieved from  
<http://www.directioninformatique.com/savoir-faire-linux-obtient-une-victoire-contre-la-rrq/10573>

cases outside SFL. This approach helped me effectively deploy snowballing technique. For instance, some of the OSS developers who have had a strong relationship with communities of OSS developers referred me to the leaders of these communities to conduct in-depth interviews and case studies. Similarly, managers and department heads could connect me with enterprise clients that could vary in terms of scope of their reliance on OSS solutions; their level of involvement with OSS communities; and their level of capabilities to engage in shared OSS R&D process.

Table 4.1 shows the list of firms participated in this study and Table 4.2 shows the list of OSS communities approached and studied in this research. Overall, 40 formal interviews have been conducted with each interviewee having an average 15 years of OSS-related experience.

Table 4.1. List of Participating Firms

Case- of firm	No. of Intw.	Position	Industry	Size (Emple yee)	Scope of specialty	HQ	Year	Ownership
1 A	1	Founder/ CEO	Internet	11-50	OSSD with Drupal	Ottawa	1999	Privately held
2 B	1	Senior architect	Telecommuni- cations	10,001 +	Internet and phone	Canada	1880	Public company
3 C	1	Senior manager of product development/ Member of R&D team/ liaison between R&D team and marketing department	Computer software	5001-10,000	Enterprise information management	Waterloo	1991	Public company
4 D	1	Founder/ CEO	OSS	1-10	OS web conferencing system	Ottawa	2007	Privately held
5 E	1	Project manager	Financial services	1001-5000	Insurance	Quebec city	1984	Privately held
6 F	1	Division head	Transportation	5001-10,000	Public transport, autobus, métro	Montreal	1861	Gov. Agency
7 G	1	Director of technology strategy	Telecommuni- cations	1001-5000	Internet	Montreal	1956	Public company
8 H	1	Vice president	Entertainment	51-200	Music and entertainment	New York	1998	Privately held
9 I	1	Vice president	Computer software	11-50	Software development industrialization	France	2005	Privately held
10 J	1	Founder/ CEO	Computer	51-200	OS tech	UK	2005	Privately





Table 4.2. List of Participating OSS Communities

	Community/ project name	No of interviews	Position	Technology(ies)	Inception	Age year	Origin
1	BigBlueButton (BBB)	1	Leader	Web Conferencing System	2007	9	Canada
2	Eclipse Foundation <sup>1</sup>	2	Board of Directors	Over 250 various OS projects	2004	12	Canada
3	Tiki Wiki CMS Groupware	1	Leader	Content Management System (CMS)	2002	14	Canada
4	Linux Trace Toolkit next generation (LTTng)	1	Developer	An OS tracing framework for Linux	2005	11	Canada
5	Tig	1	Leader	Text-mode interface for Git	2006	10	Denmark
6	Odoo (OpenERP)	1	Core Developer and Board Member	Enterprise Resource Planning, a complete suite of management applications	2002	14	Belgium
7	Puppet	1	Active developer	OS configuration management utility	2005	11	USA
8	Liferay Portal	1	Active developer	Free and OS enterprise portal software product	2000	16	USA

Note 1: Eclipse Foundation has about 250 OSS projects.

#### 4.3.6 Participant-observation

“Direct observation” is pivotal in conducting case studies, as the nature of investigation denotes: “case study should take place in the natural setting of the ‘case’” (Yin, 2009, p. 109). In this study, I use participant-observation method – as a special mode of observation (see, Yin, 2009). This method places more emphasis on researcher by turning his role from passive into an active mode. Its main strength is

that the method enables the researcher to “perceive reality from the viewpoint of someone ‘inside’ the case rather than external to it” (Yin, 2009, p. 112). For this particular purpose, I conducted an industry internship at Savoir-faire Linux, Inc.’s headquarters in Montreal for the period of one full year, part-time. This opportunity enabled me to live among the OSS developers of the largest OSS firm in Canada and interact with them on a daily basis. The method enabled me to observe the behavior of software developers up close and access to visual data that could not have been otherwise obtained. The following excerpt from my field notes illustrates my thinking during the participant observer’s data collection period.

Carefully, I peek at Alex [pseudonym], from time to time, in order to figure out why he has several cellphones on his desk. The question bugging me is: “What does he do with the three phones?” “Why is he so excited and happy at times while playing around with his phones?” “Plugging and unplugging them onto his PC... checking something on them, and then going back to work”... after observing him, unnoticeably, for a few days, I approach him on his break and pose the questions: “Alex, May I ask you a slightly stupid question? [with a humorous tone as we have already established a good rapport]” he replies, “Yes, sure.” I go on... “What do you do with these phones?” He answers, “I’m trying to write some new codes to get rid of a stupid bug, I am checking the code performance on my own phones...once I’m happy, I will share them with the community...”, I then get more curious and ask: “Are they for your own fun or will these code development and sharing benefit your professional position at company as well?”; he replies, “Both, I’m doing this task for myself [own performance and fun], but eventually I’m submitting codes under SFL [abbreviation for company Savoir-faire Linux, Inc.]’s name, so the company will gain reputation as well...it’s a win-win situation.”

As the above excerpt shows direct observation is part of participant-observation method and leads to accessing empirical evidence that can hardly be otherwise obtained. However, participant-observation is not a bed of roses. As Yin (2009, p. 113) mentions, the researcher can fall into trap of “potential biases”, “assume positions or advocacy roles”, “...become a supporter of the group or organization

being studied”, or even, “may not have sufficient time to take notes or to raise questions about events from different perspectives.”

In this study, I tried to avoid these issues as much as possible by adopting focused interviews with experts external to Savoir-faire Linux. Therefore, I could remain rather objective concerning the data I collected from inside of the company. Furthermore, as an insider who is trusted and respected, and as a researcher who has already signed confidentiality agreements, I could take pictures and record conversations quite easily. Therefore, recording events and observations was not an issue most of the times.

More specifically, I used my time at the company (as participant-observer) to build relationships and trust with individuals through social activities. For example, I attended Christmas party, joined their lunch sessions in the kitchen, played football with them after work, and hung out with the employees on different occasions, etc. These integration activities allowed me to approach the potential participants more easily and let them share their minds with me more openly.

#### 4.3.7 Background of participant-observation site

Founded by the entrepreneur Cyrille Béraud in 1999, Savoir-faire Linux (SFL), Inc. has grown into being among the largest providers of OSS solutions in Canada, and the largest and leading firm in the province of Quebec in terms of breadth (variety of offerings), depth of operations, revenues, number of employees, and R&D expenditures. SFL is headquartered in Montreal with branches in other regions of Canada (Quebec City and Toronto) as well as France (Lyon and Paris).

Growing out of a small local business within Information and Communications Technology (ICT) sector serving mainly the province of Quebec market (about 90

percent), the company has grown into a medium-sized enterprise<sup>49</sup>. Today, SFL serves different regions of Canada like Ontario and British Columbia and is capable of providing OSS-based IT solutions to international clients in the USA, France, and Africa, to name a few.

SFL has been expanding in size since its inception in terms of its number of 'full-time' employees and 'sales' volumes. According to publicly available statistics provided by Industry Canada (July 08, 2014), SFL's total sales fall between the range of \$5m and \$9,999,999; its export sales are between \$500,000 and \$999,999; and its number of paid employees is about 110. Since the beginning of this study in 2013 up until the current state, SFL has been growing steadily from having about 85 employees in 2013/2014 to well above 150 in 2016/2017. The employees (mainly 130 software engineers) form an international pool (20 plus nationalities) of highly specialized workforce having vast hands-on experience regarding different OSS projects and communities (e.g., the Linux kernel, Ring, Drupal, WordPress, Java, PHP, Liferay, Odoo, Redhat, Puppet labs, Android, etc.).

SFL's core business involves designing ISs, building IT solutions and infrastructure which are mainly based on OSS projects nested in OS developer communities. Meaning that, the company approaches clients' IT needs by promoting the OSS adoption strategies and deployment of OSS-based innovations which are developed through communities—the development process is open to those who are willing and capable of making contributions and the adoption of developed software is gratis. The key expertise of SFL is:

- (a) To identify and evaluate the suitable OS technology solution and tailor it to fit the clients' needs and wants;

---

<sup>49</sup> According to Industry Canada, companies are mainly classified into small, medium-sized and large based on the number of their paid employees as follows: 1) Small=1-99 employees; 2) medium-sized employees= 100-499; 3) large= 500 employees and more. *Source*: Industry Canada. (2015). *SME Research and Statistics*. Retrieved from <http://www.ic.gc.ca/eic/site/061.nsf/eng/Home>.

- (b) To make sure the chosen OS technology is nested in a supportive OS community undergoing a forward-looking evolutionary path;
- (c) To identify, evaluate, and justify the risks associated with making the most plausible choice;
- (d) To support migration towards adopting the chosen OS technology;
- (e) To provide on-going after-migration maintenance supports to ensure smooth running of the operations; and
- (f) To train enterprise client's IT department's staff to enable them to continue with software technology evolutionary path in connection with community of developers.

The key drivers behind SFL's expertise are three. First, SFL is home to experts who have been tightly involved in development of several OSS technologies therefore they are aware of strengths, technology potentials and vulnerabilities of these OSS projects and their dependencies. Second, SFL's management team makes sure the company remains as an integrated part of the community of developers as opposed to being just a user of a publically available OSS technology. This involves sharing codes and software developments back with the communities and maintain a symbiotic relationship. Last, SFL needs to ensure "*sustainability*" and "*interoperability*" of the OSS innovations that have been already integrated into its clients' IT infrastructure (i.e., retrospective approach) or that have the potential to become integrated into future potential clients' businesses (i.e., forward-looking approach).

The first two drivers seem to be, to a large extent, manageable by internal choices that SFL's management team makes with regards to human resource strategies (e.g., hiring engineers/developers with hands-on approach to OSS development), and allocation of internal resources to making contributions to OS communities under the

guidance of in-house R&D policies and strategies. However, the third driver is subject to a broader scope of strategy formulation that involves inter-dependencies and collaboration among other actors in OSS ecosystem. All together, these three drivers seem to influence success and sustainability of the OSSD process; yet their internal workings, interconnectedness and interdependencies are yet to be discovered and explained through appropriate theories.

#### 4.3.8 Documentation

“Except for studies of preliterate societies, documentary information is likely to be relevant to every case study topic”, according to Yin (2009, p. 101). Documents come in various forms such as letters, e-mail correspondences, notes, written reports, proposals, news clippings and articles appearing in mass media, etc. (Ibid., p. 103). In this study, I used extensive use of news articles, industry reports, government portals, and videotaped speeches on the subject of OSS phenomenon encompassing its history, related controversies, development methodology, innovation process, sociological aspects, to name a few. As a participant observer, I was able to get to know the right people and be directed toward secondary sources of data. Few video examples include:

- (a) Free software, free society by Richard Stallman (2014)<sup>50</sup> at TEDxGeneva
- (b) Interview with Theo de Raadt, OpenBSD Founder (2013)<sup>51</sup>
- (c) Freedom of thought requires free media by Eben Moglen (2012)<sup>52</sup>

---

<sup>50</sup>Stallman, R. (2014, June 12). Free software, free society [Video file]. Retrieved from [https://www.youtube.com/watch?v=Ag1AKII\\_2GM](https://www.youtube.com/watch?v=Ag1AKII_2GM)

<sup>51</sup>de Raadt, T. (2013, December 14). On OpenBSD code quality and security aspects [Video file]. Retrieved from

[https://www.youtube.com/watch?v=OXS8ljif9b8&index=3&list=FLexHbPtm6RZO9HQ\\_Riy3j8Q](https://www.youtube.com/watch?v=OXS8ljif9b8&index=3&list=FLexHbPtm6RZO9HQ_Riy3j8Q)

(d) FLOSS Weekly 146: Tiki Wiki CMS Groupware (2010)<sup>53</sup>

#### 4.3.9 Data analysis methods

In the present research, since my goal is to develop an inductive theory based upon observations from OSS industry, I adopt the data analysis approach as recommended by Strauss and Corbin (1990). This approach, according to Strauss and Corbin (1990, p. 57), aspires to: a) “build rather than only test theory”; b) give the research process the rigor necessary to make the theory “good” science”; c) help the analyst to break through biases and assumptions brought to, and that can develop during, the research process”; d) “provide the grounding, build the density, and develop the sensitivity and integration needed to generate a rich, tightly woven, explanatory theory that closely approximates the reality it represents.” Therefore, the central piece in the analysis is “coding”; the process that represents: “operations by which data are *broken down*, *conceptualized*, and *put back* [emphasis added] together in new ways. It is the central process by which theories are built from data” (Strauss & Corbin, 1990, p. 57).

Coding is the process of data analysis, and there are three major types of it: 1) *open* coding; 2) *axial* coding; 3) *selective* coding (Ibid., p. 58). Authors further highlight that the lines between these are artificial, meaning that they do not necessarily happen in stages.

*Open coding* is defined as “the process of breaking down, examining, comparing, conceptualizing, and categorizing data” (Strauss & Corbin, 1990, p. 61). In short, it

---

<sup>52</sup>Moglen, E. (2012, May 19). (c) Freedom of thought requires free media [Video file]. Retrieved from

[https://www.youtube.com/watch?v=sKOk4Y4inVY&list=FLexHbPtm6RZO9HQ\\_Riy3j8Q&index=5](https://www.youtube.com/watch?v=sKOk4Y4inVY&list=FLexHbPtm6RZO9HQ_Riy3j8Q&index=5)

<sup>53</sup>Laporte, M. (2010, December 31). FLOSS Weekly 146: Tiki Wiki CMS Groupware [Video file].

Retrieved from

[https://www.youtube.com/watch?v=Q8tjaz6CO3Q&index=2&list=FLexHbPtm6RZO9HQ\\_Riy3j8Q](https://www.youtube.com/watch?v=Q8tjaz6CO3Q&index=2&list=FLexHbPtm6RZO9HQ_Riy3j8Q)

refers to “the naming and categorizing of the phenomena through close examination of data” (Ibid., p. 62). Open coding is also the analytic process through which “concepts are identified and developed in terms of their properties and dimensions”; these concepts are therefore the “building blocks of theory” (Ibid., p. 74). Table 4.3 provides an illustration of open coding and categorizing.

Within open coding process, “*making of comparisons*” and “*asking of questions*” are among the main procedures to follow (Strauss & Corbin, 1990, p. 62), and it is the reason grounded theory approach (GTA) is also often referred to as “the constant comparative method of analysis” (Glaser & Strauss, 1967; pp. 101-116). We ask questions like “*What is this?*” or “*What does it represent?*” about an incident, a phenomenon, or an idea to conceptualize our data. This means that we give them names or labels (i.e., the process of “*labelling phenomena*”), and as we move forward in the analysis we can assign similar labels to similar phenomena so that we are able to compare and contrast them (see, Strauss & Corbin, 1990, p. 63). These labels are also perceived as “conceptual labels” that lead to forming “concepts”.

As the number of concepts increases, the researcher must group or “categorize” them. Strauss and Corbin (1990, p. 65, emphasis added) define *categorizing* as “*the process of grouping concepts that seem to pertain to the same phenomena*”. Each category is entitled to receive a “conceptual name” which needs to be *higher* in its abstraction level when compared to the names given to the concepts grouped under the broader concept (Strauss & Corbin, 1990, p. 65).

Categorizing is central to conducting grounded theory-based research, as the authors claim GTA deals with more than just coding and labeling, it also involves *analytically developing categories* which is *beyond listing of concepts* or even just



grouping them (Strauss & Corbin, 1990). Therefore, the researcher is responsible for choosing the categories, as they claim.

“This is where most names come from-YOU! The name you choose is usually the one that seems most logically related to the data it represents, and should be graphically related to remind you quickly of its referent” (Strauss & Corbin, p. 67).

A good rule of thumb that I follow is to look for the *logical fit* and *relevance* between and among codes. If a group or string of codes are well related (referring to or talking about the same incident), and they fit one another logically, then it makes sense for me to form a category including those concepts. In addition, if between the logically fit bags of concepts, I discern inter-relationships, then these categories or sub-categories are related to one another. Furthermore, developing a category involves determining its properties which can be then dimensionalized. *Properties* are “attributes or characteristics pertaining to a category” (Strauss & Corbin, p. 61). *Dimensions* refer to “location of properties along a continuum” (Strauss & Corbin, p. 61).

Table 4.3. An illustration of Open Coding from Field Notes, Observations, and Interview Transcriptions

Excerpt	Sample of codes generated (underlined)
My focus was to clean a bit the code and push it back to community. If you want some code to be integrated in Linux kernel, you have to make it the way they want. So the code has to be really clean and organized the way they want. You can develop code in many ways but not all the ways are clean and acceptable by Linux kernel. Also provided some new versions of the patch.	<ol style="list-style-type: none"> <li>1. <u>Sharing</u> codes, efforts, knowledge</li> <li>2. <u>Integration</u> of codes</li> <li>3. <u>Conformity</u> to community standards</li> <li>4. <u>Novelty and creativity</u> associated with <u>collaboration</u></li> </ol>
The process of code cleaning and integration is peer review process which involves revision, modification and submission of codes based on feedback provided by community in an iterative fashion. This kind of job is also called R&D and part-time took me about a year to finally integrate the code...It took seven reviews to finally get the acceptance of the community and integrate the revised version. When we developed the enhanced version of the OS then we pushed it back to the FLOSS community...If we come up with something very technically specific and of no use to community, we will not share it back to community. So we build modules that are re-usable and we have site-specific modules for which there is no point to share that code at all.	<ol style="list-style-type: none"> <li>5. <u>Iterative mode of collaboration</u></li> <li>6. <u>Feedback</u> perceived as enabler to the iterative process</li> <li>7. <u>Collaboration</u> with community perceived as real <u>R&amp;D job</u></li> <li>8. <u>Lengthy process</u> of collaborating with community developers</li> <li>9. <u>Indispensability of sharing back codes</u>; sharing perceived as part of the job</li> <li>10. <u>Re-useable vs. site specific modules</u></li> </ol>
Because what happens is that by giving back what we've done, we just put the name [of the OSS firm] online. Sure a company which uses this platform [TS-5500] and wants well-done support they can access the program for free but what happens often is that often companies do not have the know-how to take it, put it on the board, and make necessary customizations which are often required.....So often they see online that our company has done this kind of support before and they're sure that we can do it so they approach us with their request....This is another approach for selling things. This is an investment we make for future. It is true that I spend a lot of time to clean the code and develop the enhanced version. But it makes the name [of the OSS firm] appear online and it signals our capability, what we are able to do. The can see our	<ol style="list-style-type: none"> <li>11. Sharing back with community builds <u>reputation</u> for OSS firm</li> <li>12. <u>Know-how perceived</u> as key to benefiting from community-based version of technology</li> <li>13. <u>Reputation</u> causes customer attraction</li> <li>14. <u>Contribution</u> leads to <u>reputation</u> which acts as an <u>advertisement</u></li> <li>15. <u>Code sharing</u> viewed as <u>investment</u></li> <li>16. <u>Code sharing</u> used as a method for <u>signaling firm capability</u></li> </ol>

work and how the code is clean. Proprietary software firms rely too much on their marketing department which in the end cannot prove to customers what they are capable of doing.

Some clients use communities and some asks us to do it....For example, client 'X' ....said we have the time you want and asked us to push it back upstream when they signed the contract. They understand pushing back to the community is a good thing... They understood the process that code get reviewed. We tell them we can write really good stuff for you but we tell them if we push it back to community it will be perfect because thousands of developers will look at the code and fix whatever they could find...They know that the review process by community adds real value to the project... And every time I saw some generic part of the work that is not customer specific I push it back to the community. They accepted it. ...when customer doesn't tell us explicitly like the case of 'X', this happens implicitly and I push the generic part back to community. In this case, the customer is only dealing with us and has no link with community. They don't touch the middle [community] directly... if it is something generic, we don't need a customer to agree, but again that depends on the kind of license we agree with them. So a license decides on every line of code. Under proprietary license, we don't have the permission to push it to community but that does not happen often.

17. Advertising Technical Capability

- 18. OSS clients have options
- 19. Client's awareness of OSS methodology
- 20. Client's collaboration with OSS firm due to their understanding of OS way.
- 21. Client's appreciation of value-added as a result of collaboration with community
- 22. Consent of client can play a role.
- 23. Existence of Disconnection or gap between client and community
- 24. The licence agreed upon with client plays role in OSS firm-community collaboration

---

As an example (see Table 4.4), I use the abstract category of [HEALTHY COMMUNITY] which I have come up with in the course of this research. This category is a broad and abstract concept that is made up of a bundle of minor interconnected concepts. Therefore, to operationalize healthy community, I need to present all other related concepts and explain their internal relationships.

Table 4.4. An Illustration of Open Coding and Categorizing

Category	Properties	Dimensional range (concerning each incident)
HEALTHY COMMUNITY	Governance	Strong.....Weak
	Diversity among developers	High.....Low
	R&D oriented	High.....Low
	Active	High.....Low
	History	Long.....Short
	Heterogeneity among users	High.....Low
	Leadership	Strong.....Weak
	Enterprises support	Strong.....Weak
	Technology's commercial viability	High.....Low
	.....	.....

Categories are especially instrumental to GTA, as they form the “basis of your theoretical sampling” (Strauss & Corbin, 1990, p. 73). This means that the researcher can have a clue on what to focus upon in the upcoming interviews or observational sites. In fact, “data collection and data analysis are tightly interwoven processes” (Ibid., 1990, p. 59). For example, while analyzing and collecting the data concurrently, I discovered the concept of “healthy community”. Consequently, I decided to use this concept as a guide or a keyword while talking with some other interviewees in order to find out if such concept really existed. It is however noteworthy to mention that not all interviewees might refer to the same concept in those exact two words. Irrespective of form, I wanted to find more about the properties and their associated dimensions. This kind of ‘detective work’ further helped me to reduce the level of abstractness of the inductively driven concepts.

*Axial coding* refers to “a set of procedures whereby data are put back together in new ways after open coding, by making connections between categories” (Strauss & Corbin, 1990, p. 96) or “by making connections between a category and its subcategories” (p. 97). Strauss and Corbin (1990) suggest a paradigm model through

which categories (i.e., phenomena) are related. This paradigm includes a set of main relationships: a) *causal conditions*; b) *context*; c) *intervening conditions*; d) *action/interactional strategies*; and e) *consequences* (Ibid., pp. 99-107). Put simply, some categories may be related to causal conditions, while some others may be classified under consequences.

For example, in this study, I discovered that ‘lack of suitable dual-purpose leadership’ can lead to ‘OSS project forking’; where the former refers to a “causal condition” while the latter is the “phenomenon” or “category” under investigation. Similarly, ‘clients’ active participation in iterative code development’ can lead to ‘sustainability of OSS development process’.

Furthermore, based on Strauss and Corbin (1990, pp. 101-104), *context* can play a role in explaining a relationship. A context “represents the specific set of properties that pertain to a phenomenon” (Ibid., p. 101). It is also, “the particular set of conditions within which the action/interactional strategies are taken to manage, handle, carryout, and respond to a specific phenomenon” (Ibid., p. 101). For example, it is important to find out about the conditions under which clients of OSS firms tend to actively participate in the process of OSS development. In addition, there are *intervening conditions* that refer to “broad and general conditions bearing upon action/interactional strategies” (Ibid., p. 103). For instance, the absorptive capacity of clients concerning OSS technology can intervene in their participation. Next, *action/interactional strategies* capture those data that are concerned with “managing, handling, carrying out, responding to a phenomenon as it exists in context or under a specific set of perceived conditions” (Ibid., p. 104). Lastly, *consequences* refer to certain outcomes of actions and interactions. Eventually, as axial coding captures the essence of relating categories and subcategories it also links properties and dimensions along with those.

*Selective coding* is the third type of coding which refers to “the process of selecting the core category [central phenomenon], systematically relating it to other categories, validating those relationships, and filling in categories that need further refinement and development” (Strauss & Corbin, 1990, p. 116). The keyword in this process is integration, “making it all come together” (Hammersley & Atkinson, 1983; as cited in Strauss & Corbin, 1990, p. 117), or “getting the story straight” (Ibid., p. 142). The axial coding serves the selective coding very well as it is the process during which the categories have been worked out concerning their salient properties, dimensions, and associated paradigmatic relationships all of which provide categories with richness and density (Ibid., p. 117). The central to selective coding is the selection of *core category* and the relating of all major categories both to it and to each other (Ibid., p. 142). If there is no single core category that says it all, then the researcher must choose a name for the central phenomenon (Strauss & Corbin, 1990). For the finale, the researcher needs to validate the developed theory against the data and this validating process completes its grounding approach (Ibid., 1990).

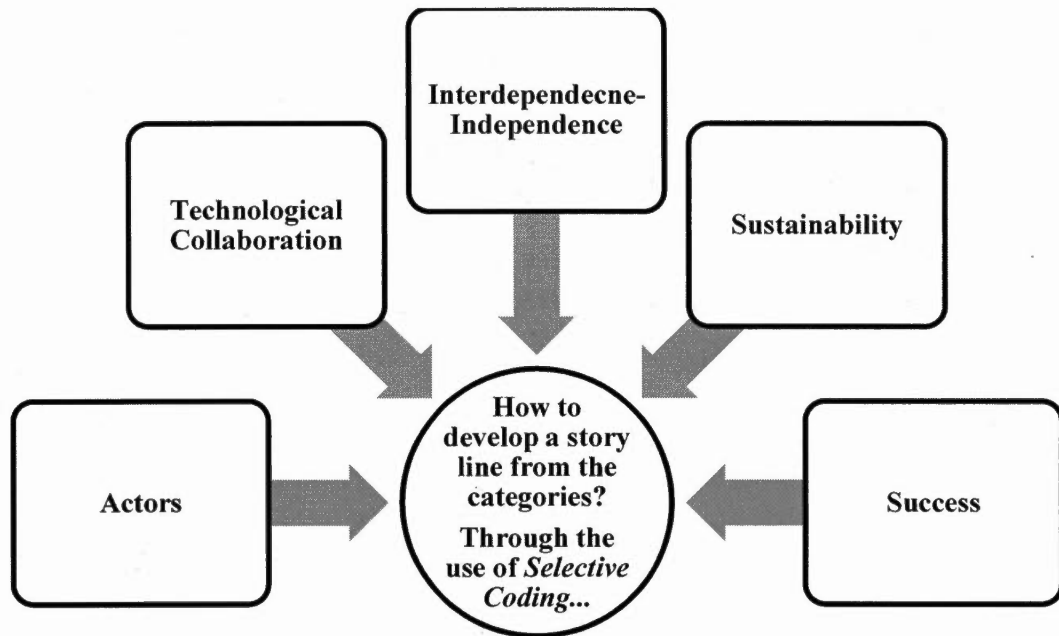
#### 4.3.10 Data analysis in the context of present study

In order to implement the GTA, I conducted a ‘*line-by-line analysis*’ of the field notes, memos and transcripts from day one the participant-observation stage and interviews began. However, sometimes when participants presented chunks of text (like a practical example) to elaborate a certain phenomenon, I decided to code the ‘chunk’ (i.e., similar to a paragraph) as a whole to represent that particular phenomenon. Overall, the open coding session was done in two stages. The first stage was concurrent with data collection phase parallel to participant observation time. Later, having finished the participant-observation phase, I decided to re-engage in the open coding activity in order to benefit from the global perspective I gained about the whole experience and the data I gathered. At this stage, open coding was coupled

with the axial coding process. My familiarity with data and the already existing codes helped me use the second stage of coding more meaningfully and effectively so much so that categories and relationships were emerging and documented all along the process of open and axial coding. At the end, the open coding process –lasting for little over a month– generated close to 360 codes– an illustration of which has been already presented in Table 4.3.

As noted earlier, a category is a class of concepts in the sense that those concepts that seem to relate to a similar phenomenon and fit logically are grouped together under a higher order– i.e., a more abstract concept called category. As the result of data analysis stage, I have discovered and formed five main categories (see Figure 4.2) each of which has their own set of subcategories and conceptual building blocks. The main categories are marked through BLOCK letters. These five main categories are: 1) ACTORS; 2) TECHNOLOGICAL COLLABORATION; 3) INTERDEPENDENCE-INDEPENDENCE; 4) SUSTAINABILITY; and 5) SUCCESS.

Figure 4.2. Main Categories Formed Through Inductive Approach



Source: Author.

#### 4.4 Research evaluation

##### 4.4.1 Validity

Yin (2011, p. 78) defines a valid study as “one that has properly collected and interpreted its data, so that the conclusions accurately reflect and represent the real world (or laboratory) that was studied”. Maxwell (2009, pp. 244-245; as cited in Yin, 2011, p. 79) further emphasizes a seven-point checklist for ensuring the validity of qualitative research. There are: 1) intensive long-term [field] involvement; 2) rich data; 3) respondent validation; 4) search for discrepant evidence and negative cases; 5) triangulation; 6) Quasi-statistics; and 7) comparison.



According to Maxwell (2009, pp. 244-245), *intensive long-term involvement* refers to producing a complete and in-depth understanding of field situations. *Rich data* encompasses detailed and varied data. *Respondent validation* means to obtain feedback from participants in order to lessen the risk of misinterpretation. *Search for discrepant evidence* involves testing rival or competing explanations. *Triangulation* is the process of collecting converging evidence from different sources. *Quasi-statistics* are the use of actual numbers instead of adjectives; and finally, *comparison* refers to explicitly comparing the results across different settings, groups or events.

In this study, I have spent a year (part-time) in the largest OSS firm in Canada and have visited other OSS firms, clients' enterprises, sat with several leaders of OSS firms, and have engaged in formal and informal conversations with managers, engineers, policy makers in order to gain different perspectives and compare and contrast opinions on similar phenomena. The duration of my field study and my intensive interaction with the context, in which the OSSTCs have been embedded, enhance the validity of this research. Furthermore, as I have built a good relationship with most of the participants and respondents, I could easily ask them to go through the transcribed interviews and alleviate any potential source of confusion in the documented data. Similarly, I could easily approach the respondents and ask them follow-up questions in case I was unsure of my understanding of some parts or in case I needed further explanations on specific points or concepts. The following excerpts from the transcripts provide an illustration of the respondent validation effort.

Figure 4.3. An Illustration of Respondent Validation: Excerpts from Interview Transcripts

With SFL phone we can maintain OS client base as within open source community it is important to have a phone running on open source-based distribution. Like Ubuntu which is a distribution based on Linux kernel.

[NOTE: SFL phone version which will work on Mac iOS and Windows will be worked on in near future!!!! According to Emmanuel] - Confirmed

**Interviewer**

How do you see the interactions between OSS firms and proprietary firms?

Is there a connection between the two?

Do you see a connection?

Is there a conflict between the two types of firms?

Do we need proprietary firms at all?

**Interviewee**

I think they can work together. ~~They (proprietary and open source) respond to different needs.~~ I rather prefer the proprietary software firms did not exist any more because they do not offer optimal solutions. If you constrain the access to the product's source code you will lose the efficiency. You exclude many who can help. But I do not know if it is possible to wipe out proprietary model.

~~But your experience can be great for others too: the way you architecture your configuration and this can be great for the community.~~

~~For example, I used a specific template configuration or module in my solution. I can give this hint to the community.~~

~~But even here, there are some experience and tricks like the way you architecture your configuration and this can be great for the community. Like ok I used this template and if you install this and that modules then you already have all what you need to do your configurations for this kind of server.~~

**Interviewee 2 (Thibault)**

The first reason that we do not release something is that it is not useful for the community. And the secrecy is the second one.

Respondent validation also provides interviewees with the opportunity to take back words and phrases they may not have wanted or intended to say in the first place. This approach to data validation tends to enhance integrity as well as confidentiality of corporate-sensitive issues.

In addition, I aimed to triangulate my findings. Triangulation captures the basic logic reflected by the principles of geometry in that multiple viewpoints allow for greater accuracy. Denzin (1978, p. 291) defines triangulation as “the combination of methodologies in the study of the same phenomenon”. His perspective on triangulation refers to a “between (or across) methods” (Ibid., p. 302) type which is the most popular use of triangulation (Jick, 1979). In the course of the present study, I used *direct observation* (through participant-observation method), *interviewing*, and *documentation* in order to collect data. Further, I tried to include a variety of participants in the sample in order to increase its *heterogeneity*. Interviewing participants with different educational backgrounds, various ethnicities, miscellaneous industry experiences, offers me the chance to access a variety of opinions and the possibility to conduct corroboration. As Jick (1979, p. 602) claims: “It [triangulation between methods] is largely a vehicle for cross validation when two or more distinct methods are found to be congruent and yield comparable data”. Jick (1979, p. 602) further gives the example of possibility of studying the effectiveness of a leader through observing his behavior, interviewing him, and evaluating his performance records. In line with Jick’s (1979) study, this research has built in mechanisms to ensure benefiting from triangulation technique.

#### 4.4.2 Generalizability or external validity

Generalizability refers to “the validity of a theory in a setting different from the one where it was empirically tested and confirmed” (Lee & Baskerville, 2003, p. 221). The concept of drawing generalizations based on research results has been “a major concern to those who do, and use research” (Lee & Baskerville, 2003, p. 221); however, it needs to be approached differently within qualitative (inductive or emic approach) and quantitative (deductive or etic approach) methodologies. In line with this argument, Yin (2009, p. 38; see also Yin, 2011, pp. 98-102) mentions that:

“A fatal flaw in doing case studies is to conceive of statistical generalization as the method of generalizing the results of your case study. This is mainly because your cases are not ‘sampling units’ and should not be chosen for this reason” (Yin, 2009, p. 38).

Following Yin’ (2009, 2011)’s view on generalization, I can briefly touch upon two types of it: a) *statistical* generalizations; and b) *analytical* generalizations. Statistical generalization is a more commonly recognized one (Yin, 2009); yet the one which has been also oftentimes not well understood by researchers and IS researchers in particular (see Lee & Baskerville, 2003).

In *statistical* generalization, “an inference is made about a population (or universe) on the basis of empirical data collected about a sample from that universe” (Yin, 2009, p. 38). Therefore, the researcher tries to establish a relationship between the research sample and the population it represents based on “numeric estimates” (Yin, 2011, p. 99). This is the common method of generalizing while conducting survey research (Fowler, 1988; Yin, 2009; Yin, 2011). Emphasizing generalizability is mainly a “key feature of the philosophical tradition of positivism” (Lee & Baskerville, 2003, p. 229) where its “sole aim is to discover invariable universal laws governing phenomena” and it “seeks to determine the universal laws governing every observed phenomenon” (Kolakowski, 1968, pp. 58-59).

According to Yin (2011, p. 98) particularistic nature of qualitative research which denotes “understanding nuances and patterns of social behavior” demands the researcher to study “specific situations and people complemented by attending carefully to specific contextual conditions”. Lee and Baskerville (2003, p. 231) claim, “a typical and legitimate endeavor in interpretive research is the study of a single setting”. Therefore, in view of “particularistic feature” of qualitative research it is difficult to consider how findings can extend beyond the immediate study, and

become generalized to some broader set of conditions (Yin, 2011, p. 98). After all, Campbell and Stanley (1963), inspired by Hume's truism (1993), posit that "*induction or generalization is never fully justified logically*" and that "we cannot generalize at all" (as cited in Lee & Baskerville, 2003, p. 225).

However, a qualitative researcher, who conducts a case study, needs to aim for "*analytic generalization*" and "avoid thinking in such confusing terms as 'the sample of cases' or the 'small sample size of cases', as if a single-case study were like a single respondent in a survey or a single subject in an experiment" (Yin, 2009, p. 39). Put simply, case study is concerned with *analytic generalization* in which scenario the researcher "is striving to *generalize a particular set of results to some broader theory*" (Yin, 2009, p. 43, emphasis added).

Lee and Baskerville (2003) identify 'generalizing from description to theory' as a valid mode of generalizing from qualitative data. This involves using empirical statements as inputs to the generalizing process in order to develop theoretical statements that are considered as output of generalization (Lee & Baskerville, 2003). Walsham (1995a, pp. 70-80) explains that a researcher can start with facts or rich descriptions of a case and then move on to generalizing to concepts, to a theory, and also to specific implications, or to rich insights. Furthermore, Lee and Baskerville (2003) draw on the tenets of GTA (Glaser & Strauss, 1967) such as building a theory deeply rooted in descriptive categories, and the emerging relationships among them based on properly collected and coded data. They take this approach to further note that "this squarely fits the phrase 'generalizing to theory' and the phrase 'generalizing from empirical statements to theoretical statements'" (Lee & Baskerville, 2003, p. 237). Lastly, Eisenhardt (1989, pp. 546-547) emphasizes the strength of developing theory from cases. She highlights the "likelihood of generating novel theory", the likelihood of building a theory that "is likely to be testable with constructs that can be

readily measured and hypotheses that can be proven false”, and finally she mentions the likelihood that “the resultant theory is likely to be empirically valid” (Ibid., pp. 546-547).

In this study, I follow three variations of qualitative methodology; namely, case study, participant-observation and GTA. I, therefore, aim for *analytic generalization* which fits case study methodology and theory building approach of grounded theory. This mode of generalization is one that is “commonly practiced but not commonly recognized” (Yin, 2011, p. 99).

#### 4.4.3 Reliability and ethical considerations

Reliability means, “if a later investigator followed the same procedures as described by an earlier investigator and conducted the same case study all over again, the later investigator should arrive at the same findings and conclusions” (Yin, 2009, p. 45). Its goal is to reduce the errors and biases in the research (Yin, 2009).

Yin (2009) proposes two techniques: 1) “case study protocol”; and 2) “case study database”. A researcher can use these tools to enhance the reliability of his case study research. To enhance the reliability of this research, I used a detailed and comprehensive ‘*research protocol*’ and created a ‘*research database*’.

My research protocol includes three main stages. First stage includes preparing the ‘*research proposal*’; the second is undergoing the ‘*ethics training*’ related to research which involves human subjects; and the third is applying to the university’s ethics board (i.e., Comité institutionnel d’éthique de la recherche avec des êtres humains: CIEREH, UQAM) in order to receive the *ethics approval certificate*.

On a more detailed note, the research proposal includes all mainstream major parts such as introduction, problem statement, objectives, literature review analysis, preliminary research questions, research logistics (time, tasks, and cost scheduling), research methodology (theoretical approach, data collection methods and data analysis techniques), and proposed theoretical and practical contributions. My proposal has gone through the triple blind-review process by industry and academic experts in order to be deemed appropriate and worthy of investigation. Next, I took the research ethics' training course with the principal research investigator in order to learn about the nuts and bolts of key issues in research ethics. I provided detailed responses (in form of a paper) to the three main sets of questions:

- 1) What is the basic history that led to the development of research ethics policies of informed consent?
- 2) What is the definition of informed consent? How does a researcher know that she has obtained informed consent? (i.e. the conditions for seeking informed consent)
- 3) Your research does not involve experiments with human subjects, but rather observations. How should the standards for informed consent and seeking participation of subjects be different, if any?

Further, I embarked on making a formal application to CIEREH in order to receive the formal approval, which would further allow me to enter the site(s) of investigation and conduct the fieldwork. This application packages involves all detailed questions regarding major and minor ethical considerations of conducting fieldwork. To make sure no confusion would remain, and to err on the safe side, I also prepared an appendix to the application package detailing all nitty-gritty about the following issues:



- a) Interview procedure (preamble, administration of interviews, closure),
- b) Interview types, duration and number,
- c) How issues of anonymity and confidentiality are cared for,
- d) Data security management,
- e) Sample of interview protocol form (preamble checklist and sample interview questions, as well as closure checklist),
- f) Data transcription procedure (voice file arrangement, confidentiality and data management, transcription site),
- g) Flowchart of the interview process,
- h) Consent forms (in English and French),
- i) Non-disclosure agreement.

To further ensure reliability, I created a *research database* in both formats: electronic and hardcopy. The electronic database included audio and video data files; transcriptions; my field notes; secondary data files like news articles, documents, papers, etc.; pictorial data files which included pictures I took from focused group discussions and conversational settings in which we used the whiteboard and charts. Further, I also kept those files that were very important in duplicate. I created the hardcopy database to keep a record of them under lock and key. Having used these two techniques, I ensured reliability of this research under strict ethical considerations and standards.

#### 4.5 Methodological contribution

In consideration of aforementioned discussions, this research makes a modest methodological contribution due to its novelty to using case study approach in combination with participant-observation and GTA. Using a combination of the three variations of qualitative approach, on the one hand, and collecting a variety of data



sources, on the other, can lead to “data triangulation” as well as “methodological triangulation” (see, Yin, 2009, p. 116). All these efforts can help me more precisely capture the nuances of concepts and formulate a grounded theory of OSSTCs as they are embedded in the OSSD processes.

Others have also profoundly progressed our understanding of the OSS projects and the factors contributing to their success. As an example, Steven Weber (2004) discusses the success of OSS projects (Linux, Apache) through the lens of political economy. However, to the best of my knowledge, he does not go into detail to discuss his methodological approach or approaches that lends support to validity and reliability of his findings. In this research, I made no assumptions about the readers’ ability to discern how I arrived at the concluding remarks, claims, or explanations. In fact, the group of these efforts to create the possibility for conducting triangulation tends to positively bear upon the reliability and validity of the present study.

## CHAPTER V

### RESULTS AND DISCUSSIONS: AN INDUCTIVE THEORY OF SUSTAINABLE OSS R&D AND INNOVATION PROCESS

#### 5.1 Introduction

This chapter aims to answer the study's primary and secondary research questions (see Table 1.1.) which are related to the '*nature of OSS value creation processes*' and the '*principal actors*' involved in the production, maintenance, and commercialization of strategic OSS solutions (i.e., collaboration-driven questions). More specifically, the sections are designed so as to provide detailed explanations on '*how*' the key actors are interrelated to one another, and '*how*' their technological collaborations influence the success and sustainability of OSS R&D and innovation process. The effort to unpack the ways of collaborations and the embedded details also encompass issues related to the enterprise clients' roles and functions (i.e., market or client-driven questions). Furthermore, as the value creation process forms a major part of the open business model seesaw, as explained earlier in Chapter Two, by exploring how '*openness*' –associated with OSS development process– is exploited by different actors, we can better understand the internal workings of OSS OBMs. As a case in point, porous boundaries of OSS R&D projects enables and actually promotes '*third-partner dependencies*' so much so that some enterprise clients can heavily invest in these projects and become influential stakeholders. However, this interdependent value creation network of firms and individuals poses the challenges concerning power distribution, using different mode of relationships, creating different capabilities and the conducive context that induces further collaborations, and perhaps most importantly creating a dual-purpose leadership (i.e.,

leadership-driven questions) that ensures the technological collaborations are actually effective and goal-oriented.

Thus, having analyzed data moving from details (observations and interview details) to generalities (groups of concepts), I have then re-organized the data and arranged this chapter around the five main inductively-driven categories (actors, technological collaborations, interdependence-independence, sustainability, and success), and delved deeply into each category's subcategories, and further detailed the conceptual building blocks of each downstream subcategory. Finally, I put all these outcomes into perspective by delineating them through 'four core propositions', a 'theoretical model', followed up by discussions and testable hypotheses. These two final outcomes form the *global view* of the inductive theory I have intended to develop based on the observations made in OSS industry.

## 5.2 Concept of OSS technological collaborations

### 5.2.1 Background based on pilot interviews

During my pilot study, I conducted two rounds of pioneering interviews with Savoir-faire Linux (SFL)'s top management team; namely, vice president of technologies, and the president (software engineers with more than 15 years of experience in ICT sector and OSS industry) in order to have a preliminary understanding of constituents of OSS technological collaborations (OSSTC). Based on over three decades of experience, the president of the company has formed some ideas about the significance of including their enterprise clients into the OSS collaborative value creation process that includes OS service providers and developer communities. He views such integration advantageous for both clients, and the whole OSS community. Based on this view, he has proposed the "noeud borroméen" (i.e., "Borromean

Links”, BL) idea which is based upon the original mathematical concept of “Borromean Rings with Brunnian Property”. Borromean Rings represents a connection of three independent elements symbolized in shape of rings. In their entirety, they stay connected unless just one of the rings is severed. This formulation of the three groups inform us of the *incongruous combination of interdependencies* among the building blocks embedded in the software development value chain; namely, enterprise clients, OS service providers like Savoir-faire Linux Inc., and OS communities. Therefore, the early discussions and brainstorming sessions have led me to develop a preliminary and fuzzy understanding of constituents of OSSTC embedded in OSS RDIP—one that is yet to be deeply investigated and supported by empirical evidence. Furthermore, we do not know about the underlying building blocks of these constituents; and, how they unfold in relation to other systemic elements.

### 5.2.2 Origins and properties of “Borromean Links”

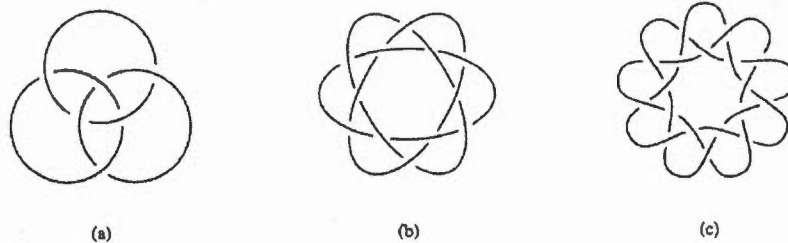
The Borromean Links (BL) first appeared in the context of mathematics in the domain of the earliest works on knots (i.e., knot theory) by Peter Tait<sup>54</sup> (Tait, 1876; See Cromwell, Beltrami, & Rampichini, 1998<sup>55</sup>). The “Borromean (or Ballantine) link” (see Figure 5.1 below), according to Liang and Mislow (1994, p. 27)<sup>56</sup>, “is among the most fascinating of topological constructions: Three mutually disjoint simple closed curves form a link, yet no two curves are linked. Thus, if any one curve is cut, the other two are free to separate”.

<sup>54</sup> Tait, P.G. (1876). On Knots. *Transactions - The Royal Society of Edinburgh*, 28, 145-190.

<sup>55</sup> Cromwell, P., Beltrami, E., & Rampichini, M. (1998). The Borromean Rings. *The Mathematical Tourist*. 20(1), 53-62.

<sup>56</sup> Liang, C., & Mislow, K. (1994). On Borromean Links. *Journal of Mathematical Chemistry*, 16, 27-35.

Figure 5.1. Reduced Diagrams of Borromean Links in Tait's Series



Note: (a) The Ballantine link, with 6 crossings; (b) The link with 12 crossings; (c) The link with 18 crossings.

Source: Adopted from Liang and Mislow (1994, p. 27)

These links which demonstrate the property of being trivial, having removed any component, were further studied by Hermann Brunn<sup>57</sup> in another early work (Brunn, 1892) where they have been dubbed “*Brunnian links*”. Although Brunn referenced the Tait's examples, none of them used the term “Borromean” (see Cromwell et al., 1998). Thus, the earliest use of the term: Borromean, found in the literature of mathematics dates back to 1962 in which case an overview of knot theory was put forward by Ralph Fox<sup>58</sup> (Cromwell et al., 1998). “On pages 131-132, Ralph Fox uses the Alexander polynomial to show that the Borromean rings are truly linked” (Cromwell et al., 1998, p. 53).

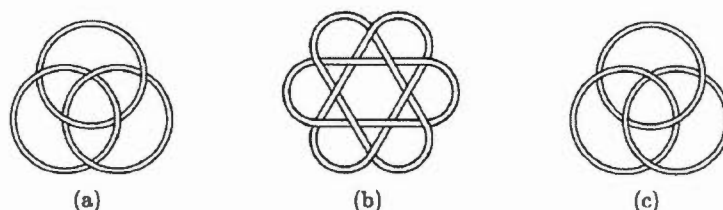
Liang and Mislow (1994, p. 28) define “A  $n$ -Borromean link as a nontrivial link in which  $n$  rings,  $n > 3$ , are combined in such a way that any two component rings form a trivial link”. They further define *ring* as “an unknotted closed (smooth or polygonal) curve” (Ibid., p. 28). Further they claim that using this definition, “all the members of Tait's series, including the Ballantine link, are 3-Borromean links” (Ibid., p. 28).

<sup>57</sup> Brunn, H. (1892). Über Verkettung. *Sitzungsberichte der Bayerischen Akad. Wiss. Math-Phys, Klasse* 22, 77-99.

<sup>58</sup> Fox, R.H. (1962). *A quick trip through knot theory*. In M. K. Fort (Ed.), *Topology of 3-manifolds and related topics* (pp. 120-167). Prentice Hall, Inc.

Cromwell et al. (1998, p. 53) mention that “Tait used the Borromean rings and another link of similar construction to show that what he called ‘belinkedness’ (now called linking number) is not sufficient to distinguish links” (see Figure 5.2). Each figure “shows an alternating, 3-component link such that each component is unknotted and no two components are linked”. However, Tait (1876) concludes that “the two links are non-trivial and different”; thereby implicitly implying the idea that “an irreducible, alternating diagram has minimal crossing number-one of the recently proved Tait conjectures<sup>59</sup>” (Cromwell et al., 1998, p. 53).

Figure 5.2. Some 3-Component Links from Tait (1876)



Source: Adopted from Cromwell et al. (1998, p. 53)

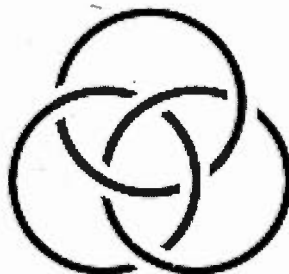
*The Borromean symbol, therefore, is based on this symmetrical arrangement of three intersecting circles. This design of three interlinked circles is known as the “Borromean Rings”. The three rings taken together are inseparable, but remove any one ring and the other two falls apart. Because of this property, they have been used in many fields as a symbol of *strength in unity* (see Cromwell et al., 1998).*

The Borromean rings’ concept, for example, has been used in the fields such as a) *Chemistry*: “Borromean DNA” (Mao, et al., 1997), “Borromean Molecules” (Chichak et al., 2004; Siegel, 2004), “Borromean Polymer Networks” (Carlucci et al., 2003); b)

<sup>59</sup> See: a) Kauffman, L.H. (1987). State models and the Jones polynomial. *Topology*, 26, 395-407; b) Murasugi, K. (1987). Jones polynomials and classical conjectures in knot theory. *Topology*, 26, 187-194; c) Thislethwaite, M. B. (1987). A spanning tree expansion of the Jones polynomial. *Topology*, 26, 297-309.

*Physics*: “Borromean Nuclei” (Austin & Bertsch, 1995; Zhukov et al., 1993), “Borromean Quantum States” (Aravind, 1997; Kauffman & Lomonaco, 2002); c) *Geometry* (Lindstrom & Zetterstrom, 1991); d) *Set Theory* (Venn, 1880); and various art and culture related fields.<sup>60</sup> In the present document, when I mention the term Borromean Links, I refer to the Borromean rings as represented in the following illustration (Figure 5.3).

Figure 5.3. An Illustration of Borromean Links/Borromean Rings



<sup>60</sup> See: 1) Mao, C., Sun, W., & Seeman, N.C. (1997). Construction of Borromean Rings from DNA. *Nature*, 386, 137-138; 2) Chichak, K. S., Cantrill, S. J., Pease, A. R., Chiu, S.-H., Cave, G.W.V., Atwood, J.L., & Stoddart, J.F. (2004). Molecular Borromean rings. *Science*, 304, 1308-1312; 3) Siegel, J.S. (2004). Chemical topology and interlocking molecules. *Science*, 304, 1256-1258; 4) Carlucci, L., Ciani, G., & Proserpio, D. M. (2003). Borromean links and other non-conventional links in polycatenated coordination polymers: Re-examination of some puzzling networks. *Cryst. Eng. Comm.* 5(47), 269-279; 5) Austin, S. M., & Bertsch, G.F. (1995). Halo nuclei. *Scientific American*, 272, 62-67; 6) Zhukova, M.V., Danilina, B.V., Fedorova, D.V., Bangb, J.M., Thompsonc, I.J., & Vaagend, J.S. (1993). Bound state properties of Borromean halo nuclei: <sup>6</sup>He and <sup>11</sup>Li. *Physics Reports*, 231(4), 151-199; 7) Aravind, P. K. (1997). Borromean entanglement of the GHZ state. Quantum Potentiality, Entanglement and Passion-at-a-Distance: Essays for Abner Shimony, eds. R. S. Cohen, M. Horne and J. Stachel, Kluwer, Dordrecht, pp.53-59; 8) Kauffman, L. H., & Lomonaco, S. J. (2002). Quantum entanglement and topological entanglement. *New Journal of Physics*, 4, 73.1-73.18; 9) Lindstrom, B., & Zetterstrom, H.-O. (1991). Borromean circles are impossible. *Amer. Math. Monthly*, 98(4), 340-341; 10) Venn, J. (1880). On the diagrammatic and mechanical representation of propositions and reasonings. *The London, Edinburgh and Dublin philosophical magazine and journal of science*. July – December.

### 5.2.3 OSSTC: Advantages, vulnerabilities and the relevance of the “Borromean Links”

OS software developers, in general, view OSS development process as an “inclusionary” process of software development in that anyone who is capable and willing has the opportunity to voluntarily become part of the research and development (R&D) as well as the innovation process (RDIP). They believe, OS way is a powerful software development methodology primarily because it is *efficient*, *innovative* and *effective*.

It is *efficient* mainly because developers can draw on a wide range of already-existing technology packages and libraries in order to avoid reinventing the wheel. By doing so, the community of developers as a whole saves a lot of time and much re-development costs. It is also efficient because the technical problem solving process is not limited to a restricted pond of developers as is the case with proprietary software developers. Therefore, the chances of finding suitable solutions to newly emerging problems in a relatively shorter period are far more. Next, OS way is also highly *innovative* in several ways<sup>61</sup>. Developers do not always need to introduce technical novelties, rather they can reapply the existing technologies in new combinations and configurations to solve on-going problems and satisfy their needs. Further, OS way is very *effective* in that its application in customers’ market is highly goal driven. Meaning that the real successful OSS solutions are those who fill up a gap and answer to a real customer need. The following quotes from the interviewees shed light on this issue.

---

<sup>61</sup> This view of innovation is consistent with Schumpeter’s thesis that innovation does not necessarily mean introduction of technical novelties, but reapplying the existing product designs and production methods in new market context also qualify for entrepreneurial acts of innovation (Schumpeter, 1934).



*"You have a huge base of work to build up on; you are not starting from scratch. There is a lot of tendency to reinvent the wheel in proprietary while there are more tendencies to reuse existing codes and improve it and contribute back; hopefully..."*

(OSS developer/Maintainer)

*"I am following the developer's mailing list for the Django projects which is the platform that we use here at work; and, I get to know the evolution of the project because I get to know what the developers say about the project. I get to follow the discussions about the future of the project. That is why I get to be one step ahead. Because if I follow the discussions, I get to know where the project [technology] is heading...And then if the customer needs something; for example, if he needs solutions that are made on that platform, I have this information because I have been reading it in that mailing list. By reading the mailing list activity I get to know about the internal of the technology as well."*

(OSS developer/IT consultant)

However, although the OS way offers these three advantages, its development process demonstrates three key vulnerabilities long-term; these are *maintenance*, *sustainability*, and *interoperability*. These vulnerabilities are rooted in OSS original traits: 'openness' and 'collective development nature'. OSS offers the possibility to be used literally by *anyone* gratis while drawing on collective efforts to be created, maintained and progressed. These vulnerabilities are especially critical in case of OSS as it is *software that is never finished*; i.e., OSS is always in the state of flux being evolved into a more functional, more solid and reliable software solution. The following quotes from the interviewees shed light on this issue.

*"One of the things that people seem to forget is, let's say a project is a million dollars...Ok, we have the money and do the project; but, once you deploy the new system, you need to maintain it. You need to make sure it will be kept up-to-date...it does not stop after the project is delivered. You need to have bodies to operate the system and to maintain it and so on..."*

(IT department head of publicly-held enterprise client)

To illustrate the power points and vulnerabilities of OSS and its development process, I have sketched a schema (Figure 5.4) based on the formal and informal discussions and interviews with the study participants. Figure 5.4 feeds on interactions of OSS developers (who belong to OSS firms and communities) and clients of OSS firms concerning several OSS projects (e.g., Odoo, formerly known as OpenERP, Liferay, Ring, etc.). Furthermore, Figure 5.4 is followed by Figure 5.5 that shows a microscopic view of interconnectedness among dependencies through a build-root graph of dependencies.

As illustrated in Figure 5.4 and in a more detailed form in Figure 5.5, an OSS project is founded upon a range of interconnected ‘libraries and repositories’ so much so that without these ‘dependencies’ the overall project renders ineffective.

For example, ‘Liferay Portal’, licensed under GNU Lesser General Public License (LGPL), is an OS web platform including features commonly used for development of website and portals and is written in ‘Java’ programming language. Java itself is free and OS software licensed under GPL with its core code available under free software/OS distribution terms. Therefore, Java— a 20-year old programming language— forms a foundation for Liferay and a source of strength since it is considered as one of the best programming languages with proven record of accomplishment.

To provide another example, we can look at ‘Odoo’ which is an OS enterprise resource planning (ERP) software providing potential users (i.e., mainly enterprise clients) an alternative to its counterpart proprietary ERPs<sup>62</sup>. Odoo comprises three types of modules: a) *Core* or *basic* modules; b) *Generic* modules; and c) *customized* modules. The *core* modules (e.g., base, account, sales, purchase, about 600 or so)

---

<sup>62</sup> Examples include SAP ERP, Oracle E-Business Suite, Microsoft Dynamics, NetSuite, as well as its counterpart from OS world such as ADempiere, Compiere, OFBiz, Openbravo ERP, to name a few.

have been originally created by the main OS enterprise including the core developers of Odoo project. These core modules are also being revised, further improved upon, and maintained mainly by the main OS enterprise behind the project but the company also includes the network of Odoo partners<sup>63</sup> in Odoo RDIP. The *generic* modules (e.g., 'l10n\_ca', 'purchase\_landed\_cost', 'magentoerpconnect', about 3000 or so) are mainly created through the developer community around Odoo. This community is comprised mainly of a network of partners. Lastly, the *customized* modules (e.g., 'my\_report') are specifically developed by each OSS firm based on the request of their clients. The customized layer of Odoo project is licensed under proprietary terms as their development process is financed by clients and these modules are critical part of clients' IS. The following quote is about the customized modules.

*"This [a customized module] gives the client the competitive advantage and this module will not be shared among the Partner Network sphere. Sometimes client itself builds this customized sensitive module."*

(Odoo community practice leader/department head)

Although within the professional conduct, the proprietary rights concerning customized modules are reserved for clients, this does not mean that those modules, technologically speaking, do not rely on core and general ones. Put simply, for customized versions to run smoothly and be fully effective, they need to be integrated into updated versions of their underlying dependencies; i.e., core and general modules.

As shown in Figure 5.4, the customized modules' level is built upon the basic and general modules' level, further supported by core code repositories and libraries (i.e., dependencies). Therefore, if the evolution path of customized modules parts with that

---

<sup>63</sup> Partners include OSS firms and IT consultants that depend on Odoo as one of their main services they offer.

of their underlying dependencies through time, then it will become harder and harder (in terms of technical possibility and economic rational and cost efficiency) to maintain the inter-operability between customized version of modules and their underlying dependencies. Under such circumstances, the client may no longer benefit from the strengths of deploying OSS solutions as maintaining an 'orphaned' OSS without the support of communities. It becomes highly costly, ineffective and making improvements or adding new features becomes a highly redundant process<sup>64</sup>.

---

<sup>64</sup> Unless, the client deems so like in case of defense industry in which case operators only would like to download and use OSS solutions to skip some primary R&D costs, and then develop the rest of the project and maintain it in-house.

Figure 5.4. OSS Development Process: Interplay between Evolution and Inter-Operability (Compatibility)

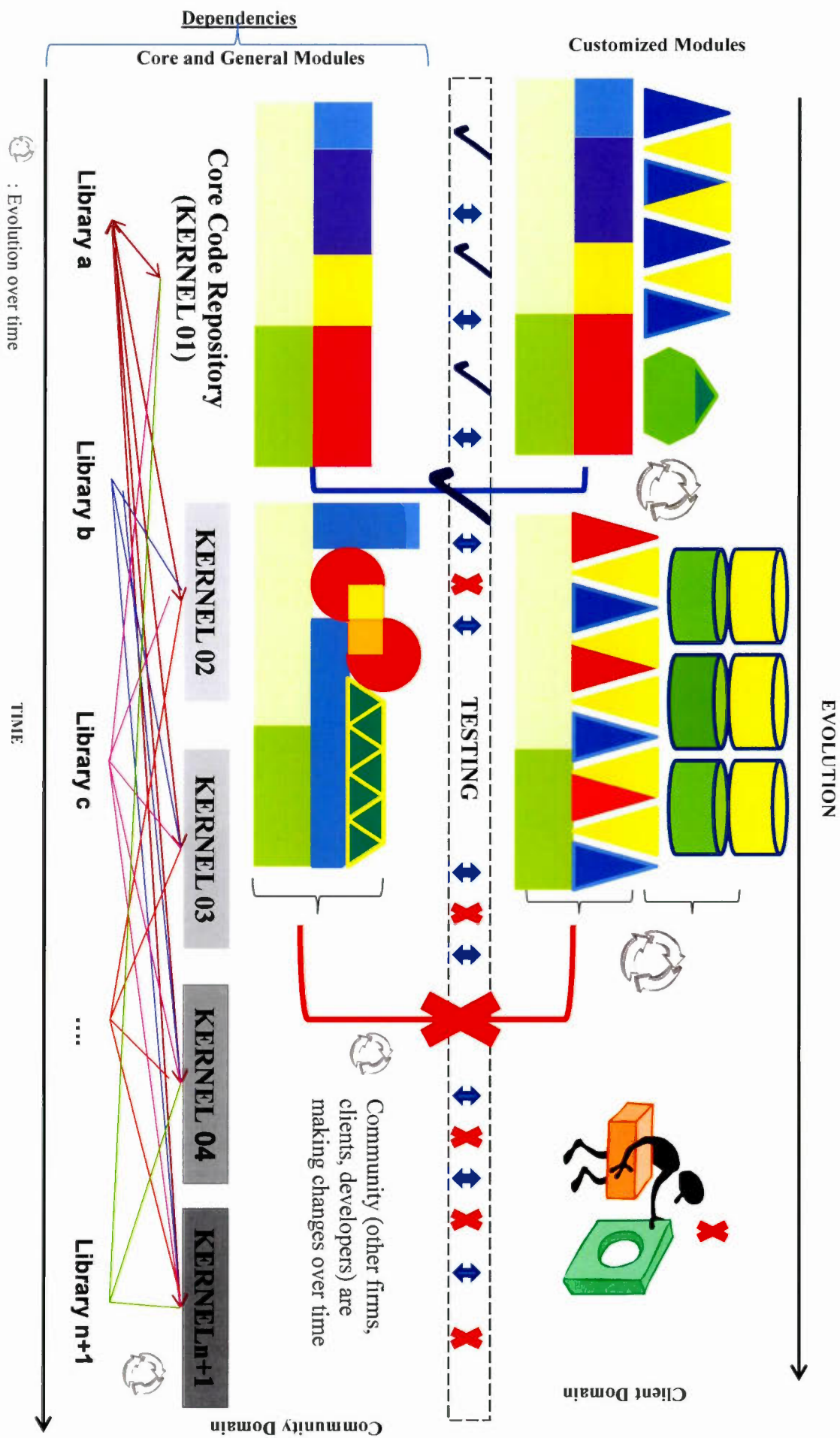
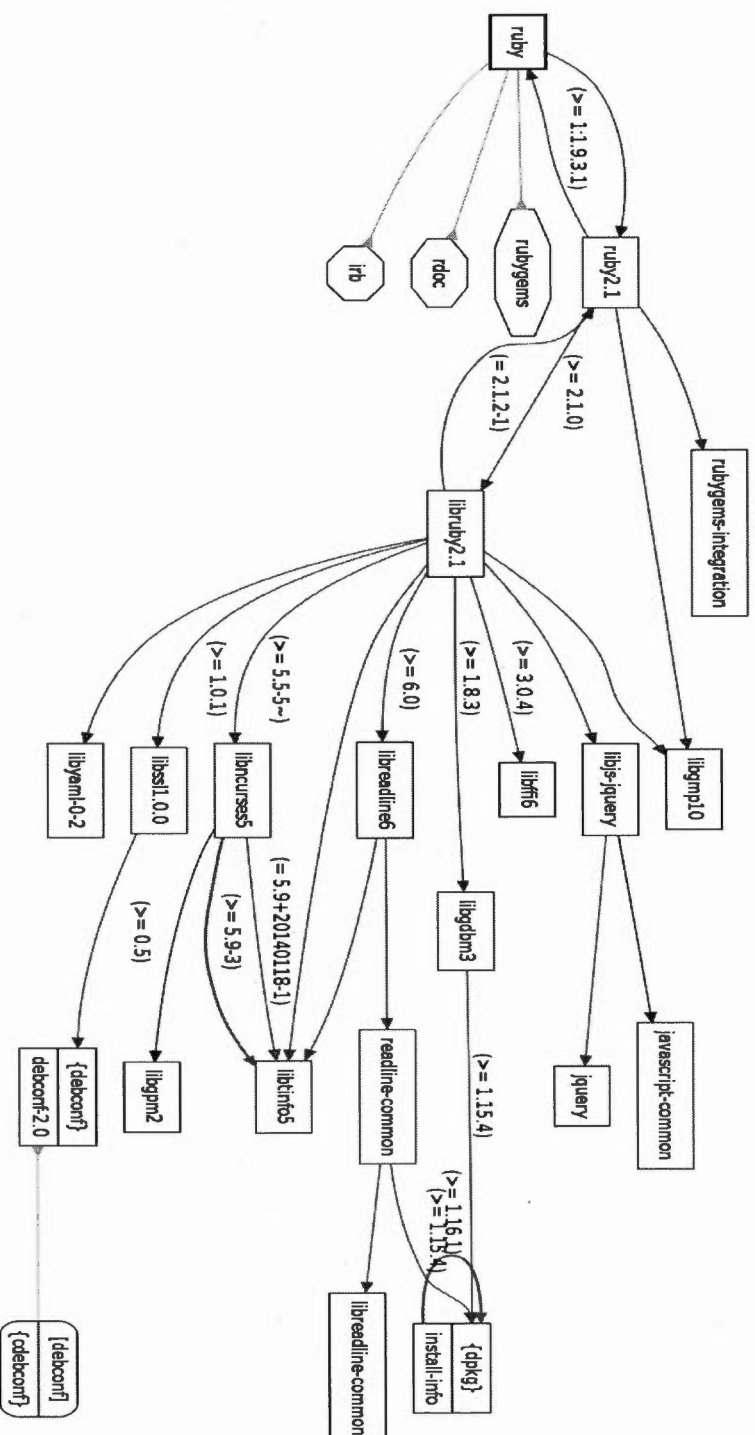


Figure 5.5. A Sample of Build-Root Graph of Software Dependencies: “The Dependency Graph of the Ruby Packages”<sup>65</sup>



<sup>65</sup> This diagram is taken from the online article: “An introduction to the Debian Continuous Integration project” published at the “Software Livre Brazil” website which can be retrieved at <http://softwarelivre.org/terceiro/blog/an-introduction-to-the-debian-continuous-integration-project>

These two quotes below further demonstrate how the enterprise clients view the significance of *software maintenance*. They reveal that getting software codes integrated into an OS platform is an achievement for enterprise clients for two reasons. First, they can save some costs of maintenance and they can subtly influence the OS technology trajectory as it bears “their” version of technology too.

*“It is true in that you can leverage the product to adapt to your context. But at the same time, when you do that you have appropriated the product for yourself and after that it’s more difficult to maintain the software in line with current developments. So you have to be very [careful]... it is important to know what you are doing!”*

(OSS enterprise client’s IT department head)

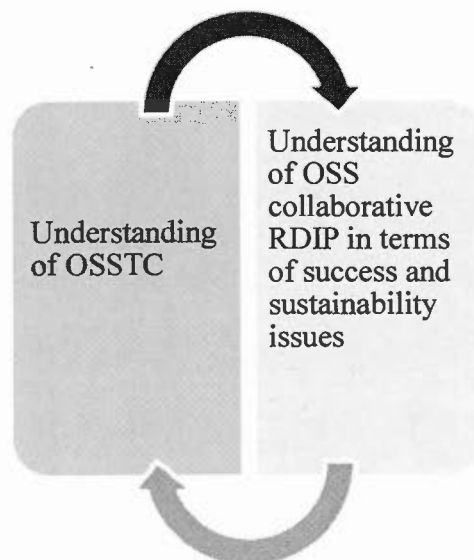
*“When it eventually gets integrated into the [OS] platform, then I don’t need to maintain it anymore...like for the next 10 years so the modules that I have added to this OSS gets integrated into the normal releases then I don’t need to maintain it anymore; because it’s either maintained by the community or the provider that’s behind it. That’s a benefit for me because it benefits the community but for me at some point these things gonna evolve on its own and I won’t need to maintain it anymore. Since we have limited resources the more others can do the better it is for us because we can work on other stuff.”*

(IT department head of publicly-held enterprise client)

In essence, OSS develops gradually and consciously in communities of OSS developers through an iterative fashion where new developments are tested and tried, revised, modified, and edited and then pushed back downstream into the underlying dependencies in order to be further integrated and maintained. Therefore, sustainability of OSS RDIP lies heavily on maintaining the link (i.e., technological collaborations) among all systemic elements in order to ensure evolution of the whole system in terms of evolution of its building blocks. This means that the sustainability of the OSS innovations is partly dependent on the existence and continuation of fruitful (i.e., goal oriented) technological collaborations among the participating

innovators. In this sense, the notion of ‘Borromean Links’– by demonstrating how interconnection among independent elements ( which are figuratively represented in form of rings) ensures unity of the whole– aptly captures the role of interconnections among OSS elements (i.e., dependencies) which ensure the unity of whole OSS as a unified system. This interconnection and interdependence further ensures the sustainability of OSS evolution process on a macro level. Therefore, the key to understanding OSS collaborative RDIP is to gain an understanding of the nature of OSS technological collaborations. More particularly, an understanding of OSSTC can also help us investigate OSS RDIP in terms of success and sustainability issues (See Figure 5.6). For instance, as the quotes show, since enterprise clients have incentives to integrate their modules into the core of the OS projects, they may have stake in their success and sustainability.

Figure 5.6. Conceptual Interdependency between OSS Technological Collaboration and OSS Collaborative R&D and Innovation Process



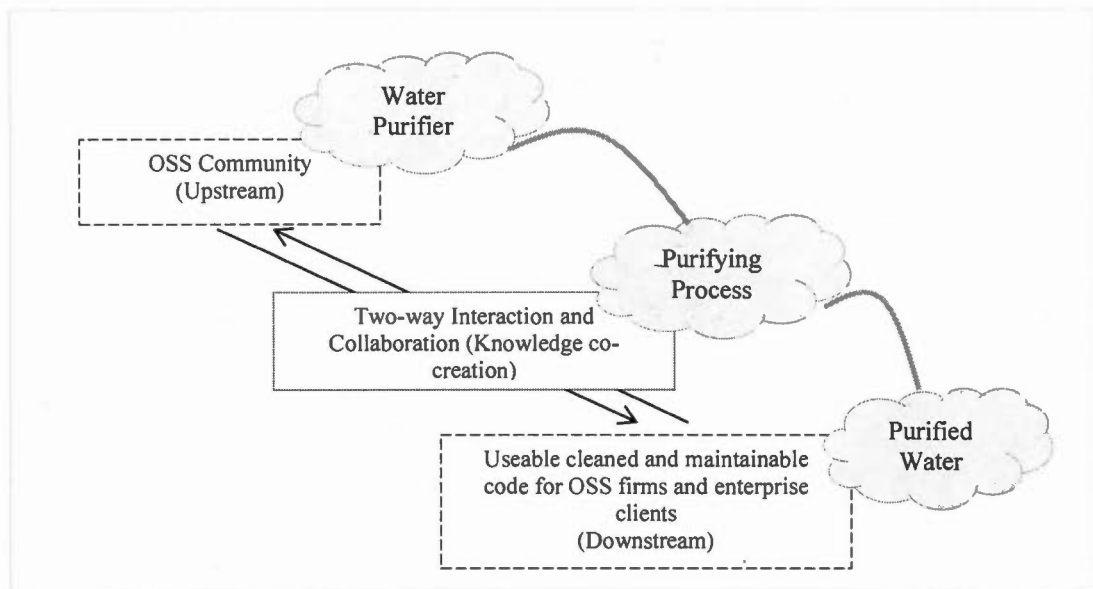


#### 5.2.4 Collaborative OSS R&D and innovation process

The collaborative OSS R&D and innovation process (RDIP) is a process that relies on a collective participation of software developers to *develop* innovations, *maintain* them and ensure their *sustainability*. Nonetheless, we know little about the details that describe the actors, the interactive process itself, the specific roles each actor plays in relation to this process, and how these roles are interrelated. To shed light on these concepts, I first touch upon the nature of OSS open and interactive process; then introduce the actors. I further specify actors' roles, the relationships, and the collaboration process as well as the leadership issues.

First, based on the bottom-up data analysis (see Chapter 4), I can describe the nature of OSS RDIP in form of a “Water Purifier Model” (WPM) where the useful knowledge needs to be co-created, constantly evolving, and flowing through an iterative and evolutionary review and development process which is open to all actors for comments, amendments, and editions. Perhaps equally important to the innovation process is the final decision making which leads to approval of a dominant design or solution (such as a piece of code written to perform a certain function, a packaging method, a patch to fix a bug). Figure 5.7 schematically represents the WPM.

Figure 5.7. Water Purifier Model: Upstream-Downstream Interaction and Collaboration among Developers



*Source:* This diagram is drawn by the author based on detailed explanation and the exact metaphor given by OSS developers.

*Note:* This concept has been emerged repeatedly in several in-depth interviews.

Figure 5.7, above, illustrates a development process which engages all actors in an open interactive manner. In this model, illustrated in form of a metaphor, OSS community is perceived as a “water purifier”, the apparatus that hosts a set of procedures that turns unusable water into drinkable water. The interactive and collaborative process that leads to knowledge co-creation is received as the “purifying process”. The outcome: The cleaned, maintainable and useable code, is recognized as the drinkable and “purified water” which benefits the OSS firms and clients downstream. An OSS developer’s quotes capture the logic behind the water purifier metaphor.

*“So by sharing code with community (upstream) the code gets reviewed like water which gets purified. Otherwise, software firm should do all the work and purify water. This is good for company and customer as both will enjoy a fine-tuned product.... It is also good for developer because now I know I can write good codes because of this because when I came here I was writing some codes as a student but it was not so good. Then I wrote some stuff and sent it upstream and the guys told me you should not do this and you should do that so I gained a lot of experience based on their feedback. That’s why I love OSS because it is also good for me and I am perfecting through my collaboration.”*

(OSS developer and consultant)

*“The process of code cleaning and integration is a peer review process which involves revision, modification and submission of codes based on feedback provided by community in an iterative fashion. This kind of job is also called R&D project and part-time took me about a year to finally integrate the code...It took seven reviews to finally get the acceptance of the community and integrate the revised version. When we developed the enhanced version of the OS then we pushed it back to the FLOSS community....”*

(OSS developer and consultant)

Through knowledge sharing and collaborative development, software developers can drive down the high cost of R&D, shorten the time to market of final products, and enhances the quality of results due to involvement of the brightest scientists and developers in the RDIP. The open and interactive process of R&D and innovation not only reflects the core principles of open innovation but also it very well resonates with the underlying logic and motivations of the firms that enter into technological or strategic alliances. Thus, by digging further into the concept we can better position OSS technological collaborations and its RDIP vis-à-vis the aforementioned two perspectives.

### 5.3 Constituents of OSS technological collaborations

In Chapter 4 (see, sections 4.3.9 and 4.3.10), I explain the data analysis methods and how I apply them in the context of this study. Based on data analysis, I categorize conceptual building blocks into sub-categories and then into main categories (see Table 5. 1). This is a bottom-up approach which help us move towards levels that are more abstract– i.e., from field observations and gathered data towards forming the broader concepts.

Table 5.1. List of the Main Categories Accompanied by Their Sub-Categories

Underlying process	Based on axial coding process	Based on open coding process
Categories	Sub-categories	Conceptual building blocks
ACTORS	OSS firms/IT consultants	Attitude and perspectives; needs and incentives to engage in collaborative R&D and innovation strategy; roles
	Enterprise clients/ Non-software enterprises or organizations	Attitude and perspectives; needs and incentives (investment incentives); key benefits; enterprise client typology (collaboration, resources, strategy, relationships); roles
	Hobbyists and individual end-users	Attitude and perspectives; roles and importance of their contribution
TECHNOLOGICAL COLLABORATION	Context for collaboration	Know-why; Know-who; Know-how; Training; Resources.
	Code sharing complexities	Licensing types; useful and general vs. non-useful and specific; secretive vs. non-secretive aspects; cost- time considerations; ethical considerations; conditions and consequences of code sharing.
	Relationships' typology	Dyadic and triadic relationships (dimensions, nature, issues, related capabilities)
	Dual-purpose leadership	Vision; setting objectives; managing technology; managing the fear of forking; managing/leading the OI community.
INTERDEPENDENCE –INDEPENDENCE	Complementarity of Roles and functions	Borromean Links representing the concept of interdependence-independence
SUSTAINABILITY (Based on selective coding process)	Technology level	Software inter-operability Software maintainability
	Resource level	Continuous tangible resource allocation Continuous intangible resource allocation
	Relationship level	Existence of dyadic relationship Existence of triadic relationship
	Leadership level	Existence of effective dual-purpose leadership
SUCCESS (Based on selective coding process)	Technological success	Quality of software code; maintainability and interoperability; quality of technology management process
	Commercial success	High commercial viability; diversity in user base; financial status (revenue model); project leadership

Source: Author.

### 5.3.1 Identifying actors and exploring the impacts of their diversity

Upon interacting with the data, and based on preliminary pilot interviews, I have identified ‘ACTOR’ (and the *heterogeneity* associated with it) as one of the key categories in the domain of OSS collaborative RDIP. The actors within the OSS innovation process have a dual role. They are perceived as both “*innovators*” and “*users*”. Based on the interviews with variety of OSS developers and users, I have developed three main sub-categories of actors who are building blocks of OSS RDIP. These three are: 1) *OSS firms or IT consultants; Enterprise clients or non-software enterprises* (including their IT departments); and 3) *Hobbyists and individual end-users* (see Figure 5.8). These three groups, together, form the concept of community of OSS developers. Had we removed OSS firms, hobbyists and clients’ IT departments from the OSS ecosystem, then the concept of community of OSS developers would have shrunk to a handful of core developers who had created or invented a technology in vacuum with little interaction with the outside world.

Since there has been growing interest in adoption of OSS solutions by many governments around the world, several interviews point towards inclusion of government bodies as users and innovators within OSS ecosystem. Further secondary sources of data<sup>66</sup>, part of which have been recommended by key informants, confirm this trend. Therefore, I tend to include government bodies within the category of enterprise clients.

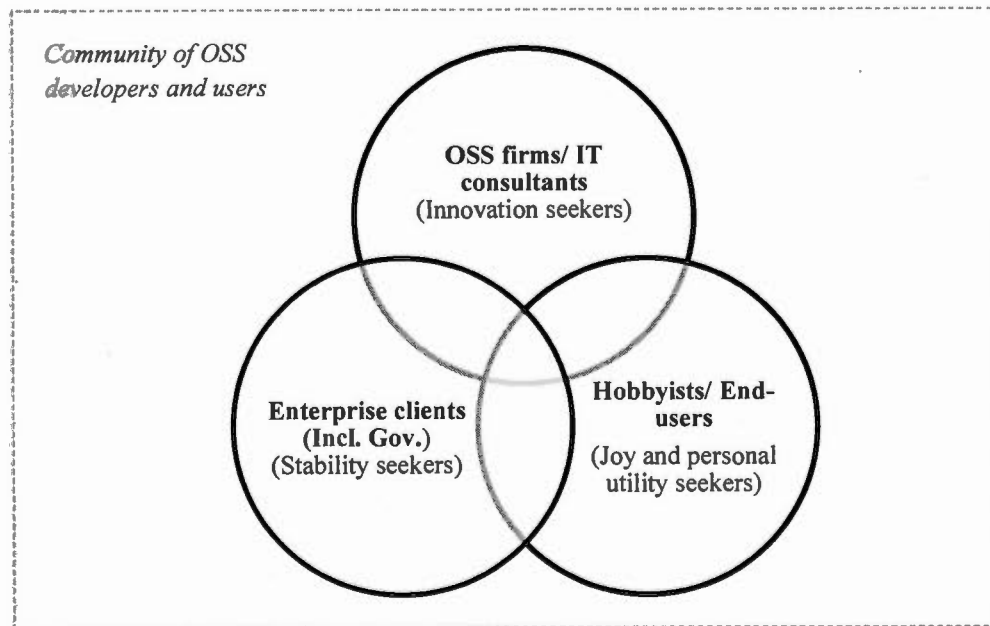
---

<sup>66</sup> Please consult the following links for further information with regards to depth and breadth of OSS adoption by different government bodies around the world: a) European Commission’s Open Source Observatory (<https://joinup.ec.europa.eu/community/osor/description>); b) Major OpenOffice.org Deployments ([https://wiki.openoffice.org/wiki/Major\\_OpenOffice.org\\_Deployments](https://wiki.openoffice.org/wiki/Major_OpenOffice.org_Deployments)). For example, the *Report on the International Status of Open Source Software 2010* shows that: “the United States, Australia and the Western European countries lead the development and adoption of open source software.” The report is available online on the website of the European Commission’s Open Source Observatory at <https://joinup.ec.europa.eu/community/epractice/document/es/gl-report-international-status-open-source-software-2010>.

The ACTOR category, therefore, forms a very abstract concept as it encompasses a variety of innovators/users who get involved in the collaborative RDIP having different *attitudes and perspectives; needs, wants, and incentives; R&D and innovation strategy (RDIS); resources; specific roles; technological capabilities; and management and leadership capabilities.*

Furthermore, as the Figure 5.8 shows, each category, in general, tends to be motivated by a major incentive. For instance, OSS firms, which are in the business of providing different services to satisfy changing needs of their clients, are more looking for innovations and innovative technological solutions, while clients are in need of stable solutions to rely on them for longer term smooth running operations.

Figure 5.8. Three Main Actor Categories of Collaborative RDIP



Source: This diagram is based on two main sources: a) the overall data analysis; and b) “Tiki Wiki CMS Groupware”<sup>67</sup> community case study

<sup>67</sup> See: <https://tiki.org/tiki-index.php>

#### A) OSS firms

OSS firms or IT consultancy firms can vary in firm size. They can be very small or micro firms with only a handful of software developers. They can also be medium-sized firms having about 100 and more full-time developers. An OSS firm can also be a very large firm like Red Hat® which is shy of 10,000 employees<sup>68</sup>. Irrespective of firm size, all OSS solutions providers share a core competency; i.e, to sell expertise on OS technologies. The following quotes bear the keyword.

*“...We sell expertise, we don't sell code!”* (OSS developer/consultant)

*“And so the idea is that... as a company we want to sell our expertise. When you work on an application like that it helps you build up your expertise...”*  
(OSS developer/consultant)

Therefore, I define an OSS firm as *an enterprise that delivers value to its clients through selling their expertise (i.e., their core competency) on adoption, customization and maintenance of OSS solutions based on what their clients' need or want.*

For example, in one of my interviews with an OS project/community administrator (i.e., key leadership position in this community), who is also an entrepreneur having his own consultancy firm, he comments about OSS firms and mentions that:

*“Their business is to use Tiki, deploy Tiki for various customers and projects and to basically deliver value to their customers. So, what typically happens is that every few weeks they have a new project which can last a year or two and they basically take Tiki and they deploy and configure it and add features to it to make their customers happy. They are pretty core [to the R&D and innovation process]. Sure, they are very core certainly to the new features that*

---

<sup>68</sup> See: [https://en.wikipedia.org/wiki/Red\\_Hat](https://en.wikipedia.org/wiki/Red_Hat)



*they add their innovation. So consultants basically using Tiki as a tool for their need and then it could be just adding more stuff to Tiki. ”*

(Tiki Wiki CMS Groupware Community project administrator)

During the interviews with OSS firms’ managers, chief technology officers, and department heads, it has become clear that these actors have a “*positive attitude*” towards playing a constructive and collaborative role in the RDIP despite having different levels of resources measured in terms of qualified labor, time, and funds. Some of the quotes shed light on this aspect.

*“We gain social proof that we are already there doing things.”*

(OSS developer/consultant)

*“That's what the open source is about: sharing codes for the benefit of others.”*

(OSS developer/consultant)

*“And we got to discuss with those guys [Liferay developers’ community] and we shared our experiences and they came up with this very generic framework for proving faceted search in Liferay.”*

(OSS developer/consultant from Liferay division)

However, OSS firms’ have a variety of “*incentives*” underlying their positive perspective to play a collaborative role in RDIP. I have investigated these incentives by not only interviewing the OSS firms’ developers and managers but also by looking at what OSS community leaders and enterprise clients view as OSS firms’ incentives. OSS firms have five main incentives, also rooted in their R&D and innovation strategy. These incentives are: 1) *Building in-house R&D technological capability*; 2) *enhancing in-house technological absorptive capacity*; 3) *signaling in-house technological capability*; 4) *branding and marketing initiatives*; and 5) *Building the necessary social capital through having a good relationship with OSS community*.

Like any enterprises in the business world, OSS firms are organizations with limited resources, as the majority of them are small and medium-sized enterprises (SMEs). However, these OSS SMEs are typically connected with OSS communities which are partly ideology-, and partly value-driven. At the same time, they are also dealing with enterprise clients which are mainly profit-driven with strict budgets, deadlines and expectations set for their IT needs. Therefore, OSS firms' decisions about making investments in collaborative RDIP are based on a variety of incentives. They must boost their in-house R&D capabilities in order to be able to remain innovative (i.e., to enhance firm-level innovation capability). Yet, their resource limitations coupled with their open approach as well as OSS technology attributes (i.e., being dependent on many other branches of software technology which exist outside the firm boundaries) do not allow them to build all technologies in-house. They need to develop R&D projects under OSS RDIP in connection with OSS world. The in-house R&D project helps them gain new technological capabilities and in parallel enhance their in-house R&D absorptive capacity through OS licensing agreements that allow them to build upon already existing projects collectively and profit from them.

In Savoir-faire Linux (SFL)— my principal case and participant observation site— for instance, the firm initiates a few of these in-house R&D projects including SFLphone (now called 'Ring')<sup>69</sup> in order to boost the company's in-house R&D and absorptive capacity in the domain of telephone or VoIP (Voice over IP) technology. The following quotes provide an insight on the nature of capacity or technological capability building through in-house efforts that are shared with OS communities.

*"He [CEO] wants to develop a home solution for VoIP (OS version). So, the technology is not developed as a request of a client. It is totally our CEO's idea. He wants to make some branding to show communities that SFL is not only client driven, but he wants to do something for community. It is a small*

---

<sup>69</sup> The official website for this technology is: <http://ring.cx/en/>.

*project but it is getting bigger and bigger. It is like an OS branding...to gain respect and build popularity."*

*"I think it [SFLPhone or Ring] was an insightful decision because there is now more demand than ever for those kind of applications. And so the idea is that, I'm guessing, as a company we want to sell our expertise. When you work on an application like that, it helps you build up your expertise of the domain of telephone and VoIP infrastructure. So, if you build this application and then in the process you really learn the domain well, and the demand for that kind of thing has gone up I believe...If the question is how you make money with that, I don't know... We contribute the most I guess."*

(Core developer of OSS firm's in-house free software R&D project)

This quote also provides an insight on the firm's incentives (e.g., technology signaling, marketing, and business development).

*"SFLphone is used for advertising and to attract people to use free software in their telephony infrastructure. So, if people [potential clients] are using a software for voice over IP phone and they find SFL phone so they would say Oh this company must know about VoIP telephony. Which we do and that's the marketing part of it. We have the packaged version of SFL phone for Debian and Ubuntu."*

(Core developer of OSS firm's in-house free software R&D project)

Further to technology-related issues, as the quotes suggest, such investments in OSS R&D projects help the OSS firm build its brand and popularity in the OSS ecosystem. These initiatives make sure the firm is an active member of OS ecosystem and is contributing to promoting OSS technology. If the R&D project becomes a popular one, the core enterprise contributor behind it will also enjoy the popularity. For instance, since October 2016, Ring has become officially become a GNU package and in 2017 it has been listed in Free Software Foundation (FSF)'s High Priority

Projects (HPP).<sup>70</sup> For an OSS firm who significantly rely on support of OS communities, these news are of strategic importance. Such efforts also lead to a very effective advertising and marketing campaign which signals how technologically capable the OSS firm is, when it comes to the technology under spotlight.

Furthermore, while interacting with OSS developers, I realized that “OSS is about people and not only the technology”, said by an OSS firm’s manager. Therefore, developing a “positive reputation” and “good relationship” with community of developers is an important part of a firm’s RDIP strategy. One CEO/entrepreneur mentions the following quote reflecting these insights.

*“Also I want to mention that today big companies have realized that it is not so much about IP but it is about people. IT technology is about the people you have, especially in the open source world. Like case of LibreOffice, I did not go out and buy the LibreOffice technology. I hired the certified core OS engineers that work on LibreOffice and that is the real asset. You have to convince people to work with you. So big companies like Google they need to go out and convince OS people to work with them. They need to be seen as good OS citizens. Red Hat and Intel are seen as good OS citizens. Google is really really bad. So, companies with bad reputation have hard time hiring OS engineers. It is important that they care about OS engineers because most of them are extremely passionate programmers and extremely good at what they do; and, in most cases, they control the projects; and, that they would like to influence further. So, these big companies are fighting over people and not technology and for that they need good reputation in the OS world.*

(CEO/Founder of an OS firm)

*“Community is about relationship and there has to be strong relationship for you and for them...Because, we know the OS community... We are working with them and we know that they are working on many different projects...like a librarian who knows where the books are...I mean; it’s a question of*

---

<sup>70</sup> See the two websites: 1) <https://blog.savoirfairelinux.com/en-ca/2016/ring-official-gnu-package/>; and 2) <https://blog.savoirfairelinux.com/en-ca/2017/ring-free-software-foundation-high-priority-projects/>

*knowledge. Like being aware of these kinds of projects, how to find them and how to use them...We speak to them and we are on the Internet with this community every day. So we are very well aware of everything that is going on."*

(CEO/Founder of OS firm)

Therefore, OSS firms' involvements in collaborative RDIP are part of their strategy to build a strong relationship with OSS communities and enhance their social capital as the result of their relationships. To build ties with community projects' members forms a strong incentive to get involved in collaborative RDIP.

OSS firms also play a unique set of roles in the collaborative RDIP. Before touching upon their roles, I present an excerpt from an interview with the R&D personnel of an OSS firm which has been the result of an entrepreneurial effort to turn an OSS project (LTTng) into a commercial product in the market place.

*"He tried at first to submit the code to the community but this did not work! Because community was not interested in his work...so Matthew moved on, on his own. He did not realize that there was a market for his project to build a company for that. That was not his first goal. His goal was to provide the tool to the community anyway possible. That was very altruistic! That's a lot work... that's the OS way! You provide value before asking for commercial side of it. So he did that project during his PhD study. After that, he wanted to continue his Ph.D. work... so why not starting my own company and be my own employee. Then some people found the project cool and useful and expressed their needs. So we provided consulting. And, that's how Efficios got started. We have a very small niche market. You need to have really special debugging needs. And we are trying to lower the barrier of entry to use our tools. So there are many tools to use for debugging performance issues and there are plenty of them available for debugging Linux. BUT, some of the problems that you need to solve will need highly specific tools like our tool. So it really depends on the kind of issues you have."*

(Project R&D and developer)

The above excerpt clarifies a major gap that persistently exists between invention or technology creation-development and successful commercialization in the OSS community projects; i.e., ‘the lack of interplay and balance between technology push and market pull of R&D and innovation’. Lack of *necessary financial resources, professional testing and quality assurance (QA), integration of end-user feedback in the R&D process* are among the key vulnerabilities of OSS projects that are developed in public domains. The following quote from BigBlueButton project leader acknowledges further these vulnerabilities.

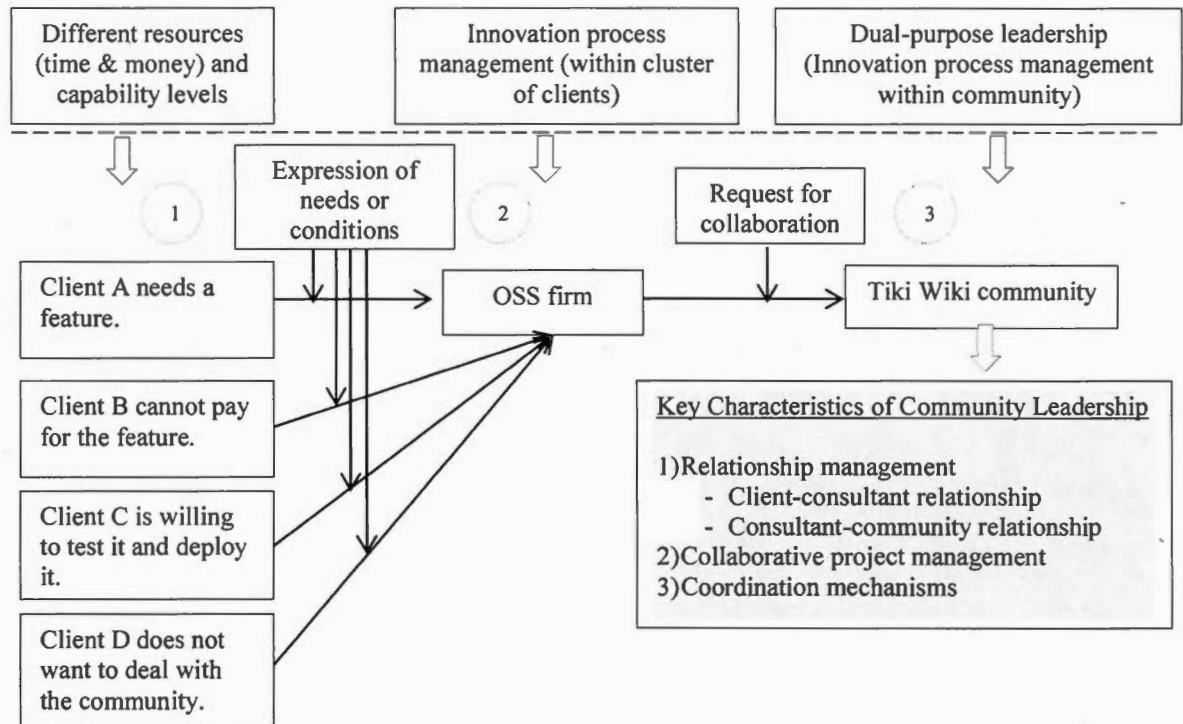
*“I would argue that all of the successful OS Projects have very strong commercial interests where the commercial company see the economic benefit of having a strong project. A lot of challenge with OSS is the quality of software itself. I call this end-user experience. It is the QA process. How can you test it? When the commercial company takes your OS project and embedded into their own product and ship to their customers, it will go through their QA teams. They will find bugs that you would never be able to find. If you make available professional services and that commercial company can engage you as one of the developers of the project, for helping them to commercialize or embed your project then not only you will earn revenue from that, and you must be able to earn some revenues from an OS project. An OS project with no revenue is not going to be viable in the long-run. A lot of times when you create an OS project from scratch, it's gonna be hard to sustain unless there is an ecosystem in which commercial companies can build upon and contribute back to the software...”*

(BigBlueButton project leader)

Therefore, based on the thick descriptions, we can clearly recognize the emerging and very specific roles of OSS firms in the collaborative RDIP. They play a significant role in terms of *filling the gap between invention and commercialization*. In fact, they provide financial support; professional testing and QA; as well as proving experienced workforce who knows both worlds: the world of enterprise clients and market needs as well as the world of passion-driven OSS developers and

hackers. Figure 5.9 provides an example of how an OSS firm engages in collaborative RDIP.

Figure 5.9. Linear Built-To-Order Collaborative R&D and Innovation Process



*Source:* Based on the interview with the project administrator of Tiki Wiki CMS Groupware community

Based on Figure 5.9, OSS firm is approached by different clients with different needs and/or conditions as well as technological capabilities. For example, one client may have enough financial resources but they cannot afford using a Beta version of OSS product. Therefore, they need to use the tested and tried version of software. Another client may be a non-for profit organization (NGO) that does not have enough budgets to pay to use a fully tested debugged version, but they agree to use the Beta version of the OS technology and help its development by going through the testing phase. Yet, another client may have good amount of end-user feedback

that can enhance the user friendliness of the finished module. In this case, OSS firm plays a key role in managing the R&D process by managing and aligning the different needs, interests and capabilities of clients harmoniously. The OSS firm can present this opportunity to the OSS community (here, e.g., Tiki Wiki community) in order to build upon the community-based version of technology and respond to individuals' needs.

In short, OSS firms have unique incentives to engage in collaborative RDIP and they benefit differently by interacting with different kinds of client organizations and software developers. Client enterprises represent different needs and they would form an important properties of OSS development and revenue model.

#### B) Enterprise clients

Enterprise clients include a variety of firms from different industries and sectors<sup>71</sup>. All of these firms have IS-related needs that can be responded to through integrating OSS solutions (tools, applications, etc.). The scope of activities of these clients is very diverse too. Some of them deploy OSS solutions to run their manufacturing operations, some others incorporate OSS into their embedded systems and ship them out to the end users, and some rely on OSS to offer online services to their end-users. Therefore, the strategic importance and commercial value of OSS technology differs moving from one client to another. One of the OSS firm's CEO interviewed in this study defines a client in the following manner. "*Client is anybody who needs software and the product you are developing, anyone*" (CEO/Founder of OSS firm).

---

<sup>71</sup> For example, in case of Savoir-faire Linux Inc., they come from aviation and aerospace, cloud computing, defense and space, electrical-electronic manufacturing, hospital equipment and services, industrial monitoring, telecommunications, financial services, IT-Software engineering, transportation, entertainment, just to name a few.



A community leader characterizes OSS enterprise clients in terms of their “IT departments” and the role they play as innovators within the OSS project’s RDIP.

*“IT departments have stuff to do. They have to make their organizations more efficient. And, that goes by providing tools to their organizations, and they have two choices either they can buy the tools or they can use open source stuff. But if you use the open source stuff the OSS stuff may not do what they exactly need so they can choose to influence the project so that’s why over a ten-year period a huge component of the contributions in Tiki comes from IT departments!... So, a client needs something, and so he writes it for themselves and then he just puts it as open-source....IT departments tend to be generally more about stability because they pick Tiki for features they had so they just want it to work! And they want to face as few problems as possible.”*

(Tiki Wiki CMS Groupware Community project administrator)

Despite their differences, clients have a variety of motivations for choosing OSS solutions. This often leads to further adoption of OSS tools as opposed to proprietary products. These incentives are expressed by them in form of benefits they enjoy through migrating towards adoption of OSS solutions.

Among key benefits clients mention are: 1) gaining flexibility associated with using OSS; 2) accessing community-based solutions to already existing problems that they are now faced with; 3) avoiding complications associated with incorporating proprietary licensed software into downstream products; 4) reducing costs associated with licensing fees and royalties; 5) benefiting from professional supports offered by OSS firms; and 6) saving on in-house R&D costs.

“*Flexibility*” is a key subcategory (concept) which means different things to different clients. It is viewed as a key benefit because it gives clients the possibility to access and use a technology that can be modified, optimized, and studied if needed.

Flexibility also refers to the OSS attribute that allows engineers to collectively develop new features and improve their quality in the public domain. All these attributes associated with flexibility are enabled by the *openness* associated and guaranteed by the OSS licensing type. Table 5.2 presents the breakdown of the flexibility concept including the relevant quotes.

In OSS world, existing answers can be obtained and applied to the relevant context without any fees being paid. Furthermore, clients can avoid patent infringements or costly licensing fees for a large number of users that are associated with embedding the proprietary licensed software into downstream products. In fact, OSS solutions can be often appropriated and incorporated into other products without paying royalty fees and signing contracts. However, this very much depends on the size of deployment. One IT department head mentions that:

*“For example, if I’m doing ECM software, for a company our size there can be .... Since we have close to 10,000 employees, the licensing fees are going to be a lot smaller than what we would pay for a comparable solution under proprietary licensing.”*

(IT department head of publicly-held enterprise client)

*“Licensing cost is very important. It doesn’t even compare. It’s a lot cheaper, initially, in regards to acquisition cost to deploy most OS solutions that it would be for something that you need to purchase like proprietary software...But what you’re not gonna pay in license fees you are going to pay in consultancy fees or deployment fees. So there’s this myth that OSS is free! Well, maybe if you use that in your basement. But for a company our size, it’s never free, I mean the deployment cost, training, all of that it’s gonna be similar and in some case more expensive than proprietary software.”*

(IT department head of publicly-held enterprise client)

Table 5.2. Unpacking the Concept of Flexibility

<i>Flexibility as...</i>	<i>Quote(s)</i>
Possibility to identify problems	<i>"...if we have some trouble with the product we can point where exactly the problem lies easily. This is in contrast with the commercial product that is closed. It is easier for us to understand the problem that is built in the product if we use OSS." (OS enterprise client IT department)</i>
Accessing community technological support	<i>"The community is often much faster and alive in responding to our problems. We like the community support. .... Based on my experience with Liferay, the community is very responsive." (OS enterprise client IT department)</i>
a) Ability to influence the technology trajectory	<i>"We can actually do changes to the system and add modules and they can be given back to the community and then they can become part of the actual solution whereas if I develop something next to Microsoft word I sincerely doubt that Microsoft will ever integrate it into their software." (IT department head of publicly-held enterprise client)</i>
b) Ability to change in the 'right' direction to create positive network externalities	
Ability to customize the software while using	<i>"On the IT side, we do provide value to actual bus users and metro users through mobile apps and websites and things like that. But on the IT side we also provide quite a lot of value to our internal customers. For example, managing the tires for the metro system; well, you can't buy a system that manages tires for the metro systems because worldwide there are probably less than 5 metro systems that use tires. So we need custom applications for things like that....So there is so much demand coming from our internal customers that we need to respond to... So when they have a problem they come see us." (IT department head of publicly-held enterprise client)</i>
	<i>"Clients can tailor free software to their needs easier. The fact that in Linux you get all these improvements for security and performance it's not magic, it's not a coincidence...having that kind of transparency, empowering people to be able to tailor the software to their needs leads to these kinds of improvements." (OSS developer)</i>
Ability to optimize internal services and products manufacturing operations through tailoring the OSS tools and solutions	<i>"Certainly, we could argue that we were able to do certain things or change certain functionalities of our system because we had access to libre software, already available technology. I'm trying to give you an example to that... using some robust and already proven...good to operate with solutions... reduce our cost of operations. The ratio of features vs. bugs or defects...because we use these already proven OS technologies, which is part of Linux kernel..." (VP of the enterprise client)</i>

Source: Author.

Thus, the costs reduction associated with adopting OSS solutions may be offset by migration and consultancy fees. However, still there is, overall, some sizeable cost saving associated with adoption of OSS solutions, claimed by our cases. One VP of enterprise client claims that *“overall, it is often less costly to acquire software if it is OSS.”* Furthermore, enterprise clients’ incentive to choose an OSS solution also is rooted in the professional support associated with that solution. A client in financial industry mentions the following quote.

*“We expect to have support for the project. So, we chose a kind of OSS project that comes with commercial support. We did not choose an OSS project that does not offer enterprise support. So for us it’s a good combination.”*

(Enterprise client IT project manager)

Lastly, enterprise clients may have internal R&D projects that can be completed with much less development costs only if they rely on some OSS projects out there. Their incentive to use OSS is mainly to avoid reinventing the wheel and continue their novelty projects privately building upon the technologies available in commons. The following quote highlights this point.

*“We have another example where we have used an OSS project but just to avoid starting from scratch so we took an OSS project and we modified the application. But we decided to use it like a start-up project and after that we used the application and leveraged it. But after that when you want to upgrade it, it is more difficult so it is important to know what you do. If you want to upgrade or adapt it is important to know that down the road having made too many changes to the original software, it would be too difficult to adapt to the community version.”*

(Enterprise client IT project manager)

However, the benefits that accrue to clients cannot be realized in vacuum; meaning that they can very much depend on certain complementary elements. For instance,

'cost saving' associated with adoption of OSS solutions is a confusing concept. On the one hand, it is very tangible financial saving up front, if software licenses do not have to be purchased. On the other hand, the clients need to undergo service support fees and training fees. Clients report that if they wish to use the OSS solutions for a long period, they need to be aware of issues associated with OSS maintenance over an extended period and take the necessary measures to ensure maintainability of their OSS tools and solutions. Therefore, as shown in a remark made by one of the clients below, a *meaningful cost saving benefit* can be realized by *remaining connected and interactive* with the community of developers.

*"A lot of times with OSS...The fact that you do have access to the source code... there may be some features that don't do what you want them to do so there's gonna be more customizations. So to me, one of the main benefits is that if you deploy OSS and remain as true to the original version as possible then the cost for a company our size [large firm] would be beneficial. For example, if I'm doing ECM software, for a company our size there can be .... Since we have close to 10,000 employees, the licensing fees are going to be a lot smaller than what we would pay for a comparable solution under proprietary licensing.*

(IT department head of publicly-held enterprise client)

*"One of the things that people seem to forget is, let's say a project is a million dollars...Ok, we have the money and do the project. But once you deploy the new system, you need to maintain it. You need to make sure it's kept up-to-date. So, there is cost to that. All the growth in the system size that you provide...it does not stop after the project is delivered. You need to have bodies to operate the system and to maintain it and so on. And now our issue is adding more people, because more people cost more money, that we don't have.*

(IT department head of publicly-held enterprise client)

Similarly, the concept of *'flexibility'* which comprises five elements, as shown in Table 5.2, again is a fuzzy and conditional concept in that a client can benefit greatly from flexibility under the condition that their IT department is connected with OSS community and collaborates technologically with the community developers. Fundamentally, issues like *community support*, *influencing technology trajectory*, *creating positive network externalities*, making and enjoying *customizations* that survive through the passage of time, and *successful optimization* that is built upon and supported by interconnect OS libraries and dependencies, all and all form the concept of flexibility. At heart of the building blocks of flexibility lie *'connection and interaction'* with OSS community of developers. The following quote elaborates on the concept of flexibility.

*"Sometimes the OSS project changes often. Its development is very fast, and sometimes it is very difficult to reuse the same application because the API has so much changed that it is more difficult to integrate it. So sometimes the OSS projects are moving so fast that it is very difficult to integrate the new version. So, it is difficult to use an OSS project because we know that it won't be the same in one year.*

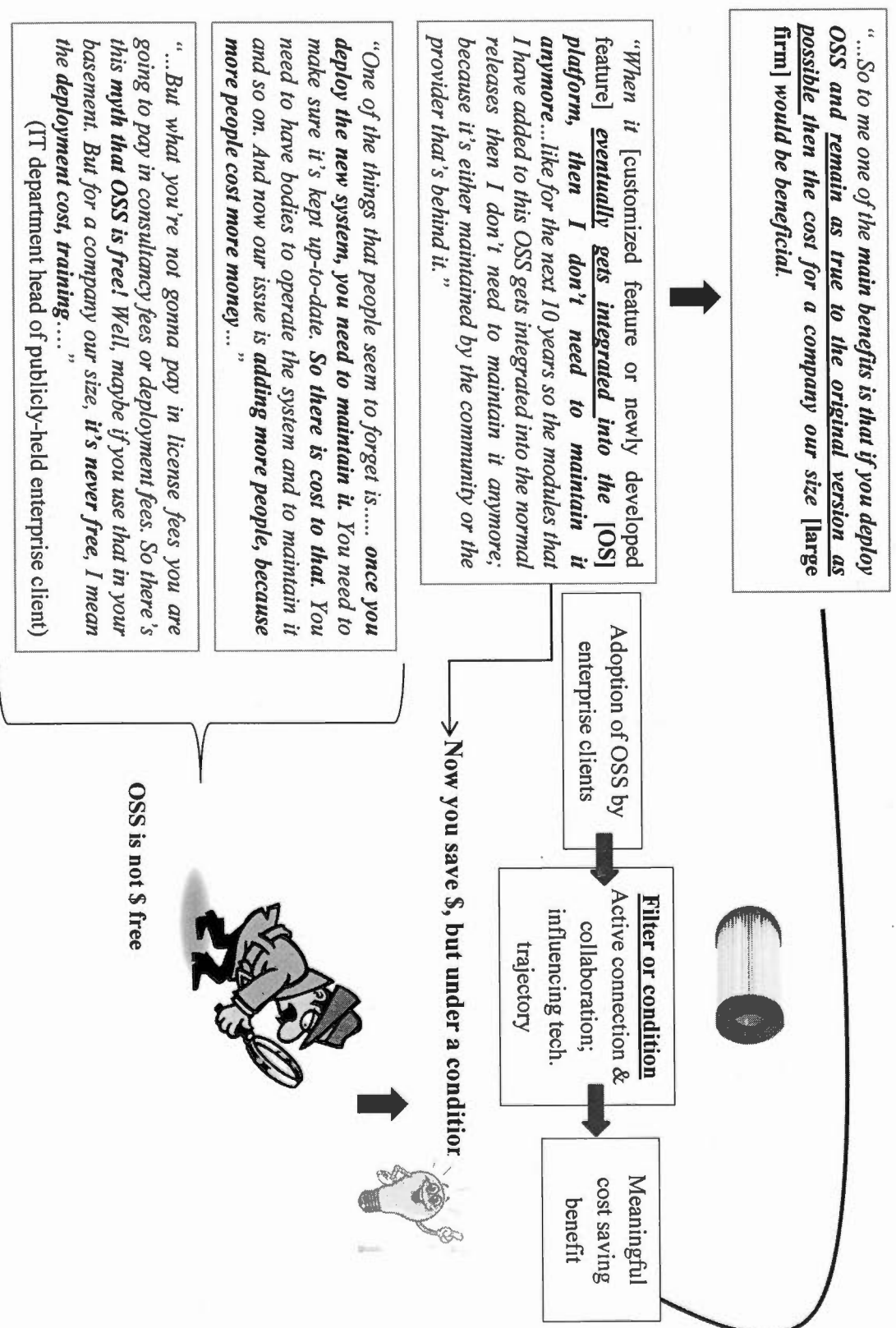
(OS enterprise client IT department)

*"When it [customized feature or newly developed feature] eventually gets integrated into the [OS] platform, then I don't need to maintain it anymore...like for the next 10 years so the modules that I have added to this OSS gets integrated into the normal releases then I don't need to maintain it anymore; because it's either maintained by the community or the provider that's behind it. That's a benefit for me because it benefits the community but for me at some point these things gonna evolve on its own and I won't need to maintain it anymore. Since we have limited resources the more others can do the better it is for us because we can work on other stuff."*

(IT department head of publicly-held enterprise client)

Apparently, benefiting from a 'rapidly evolving technology' that lives in a public domain requires the users to remain 'actively connected' with the technology source and main development platform through their technological collaborations. Remaining actively connected ensures that client' benefits (e.g., flexibility, cost saving, etc.) continue to remain as strengths of their choice of using OSS, rather than turning into vulnerabilities in their ISs. Therefore, as shown in Figure 5.10, *adoption of OSS solutions by enterprise clients can lead to obtainment of certain key benefits only under conditions of 'active connection and collaboration' (Filter or condition) with OSS community of developers.* Therefore, establishing technological collaborations do matter for clients who want to optimally benefit from their OSS adoption choices.

Figure 5.10. Relationship between Clients' Adoption of OSS Solutions and the Accrued Benefits





Having realized the intermediate role of clients' technological collaboration with the community developers of the source project puts the spotlight on the concept of COLLABORATION. In fact, the degree and mode of collaboration and connection seem to influence the proposed relationship between adoption of OSS and the accrued key benefits. Based on the nuances that differentiate clients from one another based on their technological collaboration, I propose the 'client typology' ( see, Figure 5.11).

Figure 5.11. OSS Enterprise Client Typology

Low vs. High Collaboration	Indirect vs. Direct Connection	
	QA. The Conservatives	QC. The Champions
	High indirect collaboration High indirect connection	High direct collaboration High direct connection
	QB. The Luddites	QD. The Reticent
	Low indirect collaboration Low indirect connection	Low direct collaboration Low direct connection

*Quadrant A* (QA, The Conservatives) includes clients that have a positive attitude towards adoption of OSS and OSS philosophy and ideology. They appreciate OSS development methodology and are very well aware of the fact that the promises made by this technology in form of accrued benefits are at the expense of clients' remaining connected with OSS projects and community of developers and actively collaborate in the RDIP. The IT departments of the enterprise clients that fall under this category know quite a few different OSS projects and technologies and have a general opinion about how their needs can be responded to by adopting a particular solution. However, they have neither deep expertise and technological capability nor the

necessary resources (time and labor) to directly connect and collaborate with OSS projects' community members. Therefore, they view OSS firms as both a tool to indirectly connect and collaborate with OSS communities and as a source of guarantee for support and security of their systems.

*"Sometimes we send a patch or a little new feature but directly to SFL and not the community. So we are not fully integrated into the community. We prefer to send the patch or information to our partner SFL. SFL has already the ability to communicate some new features to Liferay and they have more power to push new things like the Canadian French translation. They are already pushing the Liferay to integrate the plug-in. We try to pass it by SFL."*

(OSS enterprise client)

In addition, although these clients do not have high level of in-house OSS technological capabilities (OSS-specific absorptive capacity) which allow them to directly connect with OSS community and interact with them in R&D projects, they try to support OSS methodology and sustainability of OSS projects by indirectly investing in OSS development through their service provider. This *indirect investment* and involvement includes allowing more time to OSS firm to participate in iterative software development process with experts in community and get the codes (new modules or packaging methods) integrated into the main technology platform. Moreover, clients accept to share part of the added costs incurred by OSS firm in order to ensure that the firm assigns developers to collaborate with community on the assigned projects. In addition, the client explicitly authorizes the OSS firm on the matter of code sharing process with community. The following quotes highlight the clients' rationale for doing so, as told by OSS developers.

*"Some clients use communities and some asks us to do it....For example, client 'X' ....said we have the time you want and asked us to push it back upstream when they signed the contract. They understand pushing back to the community is a good thing... They understood the process that code gets*

*reviewed. We tell them we can write really good stuff for you but we tell them if we push it back to community it will be perfect because thousands of developers will look at the code and fix whatever they could find... They know that the review process by community adds real value to the project... And every time I saw some generic part of the work that is not customer specific I push it back to the community. They accepted it. ...In this case, the customer is only dealing with us and has no link with community. They don't touch the middle [community] directly..."*

*"... Customers must be patient and willing to support to be able to use purified water downstream."*

*"It depends on the customer type. Some customers know OSS so they understand what it is and what our task is. Some others have just a software need and they want us to respond to it. These clients do not really know about OSS communities."*

*"So the customer and community invested in the campaign with money, testing and bug report and the development was done by two partners/service providers."*

Lastly, these clients view OSS solution as a means to *gain competitive advantage* as they can work around solutions to enhance their productivity gains. Therefore, magnitude of their OSS adoption is quite significant and as they have engaged in building relationships with OSS firms, they will incur high switching costs if they decide to suddenly switch from one OSS firm service provider to another.

*Quadrant B (QB, The Luddites)* includes clients that look at OSS solutions just through the same lenses they use to consider proprietary ones. They do not have necessarily a positive attitude towards OSS and its community-based development methodology. They are mainly interested in reducing costs, and this motivates them to adopt OSS solutions if they are comparable with proprietary solutions. The benefits associated with using software that gives them flexibility do not significantly

influence their decisions. Furthermore, they face resources shortages that prevent them from collaborating even if they intended to do so. One of the clients explains that they ask:

*“Is there value for us... that’s a bigger question... certainly in this company is not clear the value of it! What would be the value of dedicating some portion of teams to contributing to OS...I don’t think that this is the business equation that has been solved yet, that anyone would be comfortable saying yes absolutely this makes sense for our business because x, y, and z.*

*“Do we have time, given the small size of this group here... It’s very difficult for us to invest...our reason of the business “raison-d’etre” is not to build software for sake of building software. We build products and services in exchange for payments for very niche markets.”*

(OSS enterprise client)

In case of the Luddites, magnitude of their OSS adoption is quite low and therefore they have very low switching costs associated with their choice of OSS adoption. Their IT departments are dominated by engineers mainly trained in proprietary software environments. Therefore, they do not have an in-house OSS technological capability that can enable them to directly connect with OSS communities. For them, OSS firms are just IT service firms that satisfy a need. Building a strong relationship with OSS firms or communities around the projects they use is not an important issue. These clients may even require strict confidentiality agreements and do not allow OSS firm to share any code back with community. Thus, when the OSS firm develops a module for them, almost all the maintenance issues become the client’s responsibility unless they sign a contract with an OSS firm for services and support. OSS firms supplying such services put it as the following.

*“What they do purchase from us are: services, training, customizations, installations, analyses for development and sometimes also we provide access to the community so we are their single point of contact with the community. So, instead of dealing with whatever the number of suppliers that you could*

*work with in a community, our customers just deal with us and we deal with the complexity of the community...* ”

(Odoo department head in OSS firm)

*“There are some who are pretty much against open source and are reluctantly using open source solutions and they use it only because it is the best tool for the job.”*

(OSS firm developer and department head)

*“We are part of the community but they also need to interact with the community.”*

(OSS firm developer and department head)

*Quadrant C (QC, The Champions)* includes clients that have a positive attitude towards adoption of OSS and free software philosophy/ideology. They appreciate OSS development methodology and are very well aware that the OSS promises in form of benefits are due to clients’ remaining connected with OSS projects and community of developers and actively collaborating in the RDIP.

*“Yes, I think there is value because the nearer you are to community, the more rapidly and easily you can find alternative solutions, push changes, make sure of your direction and use the power of community.”*

(OSS enterprise client)

*“If you want to push some change, if you are near developers it is easier... Community can help you if you have to make changes or maybe there is a work-around... Sometimes you want to know if you are in the right direction, and community can help you with this... Sometimes you need to imagine of a new solution or if there is an already existing solution for your purpose.”*

(OSS enterprise client)

The IT departments of the enterprise clients that fall under this category know quite a few different OSS projects and technologies and have a good amount of knowledge with regards to how their needs can be responded to by adopting a particular solution. However, they do not have very specific expertise, strong and close connections with

each of the OSS projects under their radar. Nor have they necessary resources like time and labor to build a *trust-based relationship* with all those communities around the projects they need. Furthermore, although they know that those projects are valuable, they need a guarantee from an OSS firm in order to justify their decision among the top managers and obtain their backing for migrations to OSS.

*“We are more driven by projects and functionality delivery....more than what’s going on out there [in communities]. Of course, when we design solutions we do a fair job of auditing what’s out there [in communities] doing a survey of what’s being used in solving these particular problems. In that sort of ...that whole reflection or thinking is associated with what X [OSS firm] can give us. OS firm for us is a pair of hands to implement a solution. We rely on their expertise to vet or influence our design or technology choices obviously; but, the value they bring to us is because they have experience with many different projects we know that from the get-go it’s going to be a high quality technology. They [OSS firms] reduce the risk factor. Without them, we have to spend more time evaluating our choices making sure we are going the right way. Their consultancy comes with a certain weight...”*

(IT department head of publicly-held enterprise client)

In addition, these clients have acceptable level of in-house OSS technological capabilities that allow them to directly connect with OSS community and interact with them in R&D projects. For example, one of the clients mentions that:

*“On our team we have people who know very well Java language so we choose an OSS which is written in Java. So our developers more easily understand the architecture or if we have some trouble with the product we can point where exactly the problem lies easily. This is in contrast with the commercial product that is closed. It is easier for us to understand the problem that is built in the product if we use OSS. ”*

(OSS enterprise client)

Although they do not have abundance of resources to build long-term and very strong relationships with communities, they have *relationship building capabilities* to remain in connection with key OSS projects' community members and engage in knowledge transfer process. The client collaborates with community through bug reporting and bug fixing (submitting patches). They also develop some features and try to get them integrated into the core project through iterative development process. The following quote sheds light on some of these remarks.

*"OS is very important for us. I think, and contrary to many other organizations, we have really positioned ourselves in the community. Any developments that we do we wanna give it back to the community. Because a lot of times companies of our size or type they're gonna take OSS, they're gonna modify it, they're gonna add to it, but they're gonna keep it to themselves. We've decided a while ago that anything that we develop that can be good for the community we're gonna give it back. For us it's important to really get into the norm of real OS process of not just taking but giving back also."*

(IT department head of publicly-held enterprise client)

*"Some clients who came to us they are already very well in contact with communities. They already began to have some contacts and ask questions and contribute back to community. But they hire us to achieve their goals faster!"*

(OSS firm department head)

*"When it [customized feature or newly developed feature] eventually gets integrated into the [OS] platform, then I don't need to maintain it anymore...like for the next 10 years so the modules that I have added to this OSS gets integrated into the normal releases then I don't need to maintain it anymore; because it's either maintained by the community or the provider that's behind it. That's a benefit for me because it benefits the community but for me at some point these things gonna evolve on its own and I won't need to maintain it anymore. Since we have limited resources the more others can do the better it is for us because we can work on other stuff."*

(IT department head of publicly-held enterprise client)

Lastly, these clients view OSS solution as a means to gain competitive advantage as they can work around solutions to enhance their productivity gains. Therefore, magnitude of their OSS adoption is quite significant and as they have engaged in building relationships with OSS firms and some community projects, they will incur high switching costs if they decide to switch from one project to another or change their OSS firm service provider.

*Quadrant D* (QD, The Reticent) involves clients that have a positive attitude towards adoption of OSS and OSS philosophy/ideology. They appreciate OSS development methodology and are very well aware of the fact that benefits associated with OSS are at the expense of clients' remaining connected with OSS projects and community of developers and actively collaborate in the RDIP.

*"Do I hope to reach that kind of level [full integration with community]?... absolutely...that would be great...there are certain areas I would love to be able to influence community in certain ways where we see greater benefits from those communities...absolutely...that's the intention...and we tend to do that through our partnership we have with SFL and other firms."*

(OSS enterprise client)

The IT departments of the enterprise clients that fall under this category know few OSS projects and technologies and have a general opinion about how their needs can be responded to by adopting OSS solutions. However, they do not have the necessary expertise or resources to choose among similar options and undergo the migration alone. As they are not well connected with OSS community projects and as they have very limited resources, they cannot afford to go through the learning curve and even commit mistakes in adoption process. Therefore, they rely on OSS firms' expertise to help them choose the right OSS solution from the "*healthy community*" and they are willing to skip the learning curve and risks of making mistakes by letting the OSS firm be in full charge of the project.



*"I think eventually over time client can go to the community and they don't need us. We see this model regularly. I mean customers coming to us for a certain level of expertise, training, etc. to get them to a certain point and then they take it back to themselves. And then at one point when they get stuck again they come to the model but this is like an in-and-out relationship it is not a consistent one."*

(OSS developer and department head)

These clients use OSS firm as a guarantor to ensure that they can get support in case they face a problem; however, they also try to remain connected with community as well and make an effort to collaborate as much as they can however little. The IT department of these enterprises includes personnel with limited OSS technological capabilities that allow them to directly approach OSS communities and interact with them in R&D projects. Yet, their expertise level is quite limited and cannot fully benefit the community knowledge repertoires. Their relationship building capability is also quite limited, as their personnel are not closely connected with OSS projects. Many may lack having deep experience working with community projects in capacities of core developers or maintainers. Irrespective of their resources and capability limitations, these clients make an effort to pose questions to community members and try to report bugs directly to community to remain connected and collaborate directly however limited. The following quotes shed some light on these remarks.

*"Well, they need us because like any other software there is a learning curve involved and even more with ERP systems... Maybe things evolve over time... Customers need us but sometimes they don't because they can manage the modifications on their own. ... Because migrating an ERP system is not like updating any other software. It is a project in itself and there is a lot of risk involved with migration. OpenERP customer would need to do a lot of debugging and scripting to have a working migrated system. Instead, you ship your database to Belgium (home to OpenERP), they do the migration and then return your new version of ERP to you. So you don't have to take care of everything. Bugfix, migration, security alerts, functional support, and*

*optional hosting if you want... Yes. We have customers going to OpenERP Apps finding modules that they want us to do the testing. We have customers on OpenERP forums asking questions to community members and they challenge our decisions and choices. We provide them with our advice. They ask questions in forum like "I have been told that we should be doing this particular business function this way..." so these presence in forum is to make sure or confirm that what they have been told to do is actually the best way to do it."*

(OSS developer and department head)

*"They don't want to spend a year and a half to be in really good contact with community to find out about the best practices and incur all the cost. For example, going in the wrong path and correct it later on. So, it is much cheaper to bring someone in like SFL with good knowledge about the best practices with the OSS tools; and to know where the tool is going; and the counseling on which feature to use and not to use. Because some features are not going to stay there long and will be disappear soon."*

(CTO of OSS firm)

*"We had on our team a gentleman who was a maintainer of an OS project...and that was great and we benefited from that and OS community benefited from that as well."*

(OSS enterprise client)

*"But we are not fully active in exchange of information with community. It needs time to participate in OSS projects. The time you need to spend with the community; to be there. So, we have to be careful how much time we are using to socialize in community, to do the social activity. For us Liferay is a product and we want to have the solution fast. You can search in community to find your answer but when you have no answer we prefer to go with our support line because we know that we would have the answer faster because they have the ability to provide the answer faster. We know that we could have the support. So relationship with community implies more time."*

(OSS enterprise client)

Lastly, these clients view OSS solution as a means to gain competitive advantage as they can work around solutions to enhance their productivity gains; but due to their resource and technological constraints they cannot be highly active with community of developers. Magnitude of their OSS adoption is quite high and as they have potentials to make strategic investments in their IT infrastructure to be more able to engaged in building relationships with OSS community projects directly and get involved with them. For these clients there is quite high switching cost if they decide to switch from one project to another or change their OSS firm service provider. Nevertheless, as they are not very much involved with communities and have not invested heavily in building relationships with OSS communities, their switching costs are lower than firms in QC.

*"We did not push back many codes; because, our main interest is to integrate the product into our systems. But, we participate in community by indicating if there is a bug in the product. So, we participate by telling them where the bug is. Sometimes we say in which class the bug is. And, sometimes we tell them where to correct the bug. But, it is very difficult or I find that very tricky to correct the bug for them (for Liferay). So, we prefer to pinpoint the problem and verify if it is really the problem or it is a design issue."*

(OSS enterprise client)

### C) Hobbyists/Enthusiasts and end-users

Hobbyists and individual end-users form the third category of actors in the OSS collaborative RDIP. Interviewees view them mainly as volunteers who get connected with projects and collaborate for non-pecuniary personal reasons including having fun, learning, satisfying some personal, technical needs, and promoting their personal ideological leanings toward free/libre software. One OSS project community leader describes this category in the following way.

*"They do open source in their spare time. They want to work on cool projects. And all that."*

(LTTng R&D project manager)

*"These are the people that are end-users of Tiki and that are not like necessarily big shot; typically people just want... one person using it for the personal or family website. And, the hobbyists are not necessarily stable because they come and go and not necessarily going to be there for about 5 or 10 years. But, there are a lot of them. And, if you add up everything they do then it's a lot of contributions. There's no real business reason for them to do it, they are not paid, and they don't really need it for themselves ...is just that some people volunteer! So what's interesting is that hobbyists because they are volunteers and sort of pick their own time, of course they are going to pick stuff which is fun for them to do, but they also going to do some stuff which is not necessarily fun!"*

(Tiki Wiki CMS Groupware Community project administrator)

*"And the other thing just come to my mind what's interesting about hobbyists is that the hobbyists or purely volunteer and they're doing everything just for the right reasons. They are not paid to do that they're doing it because it's good. And they're not under any pressure of any contracts or any deadlines or any budgets...They just do it because they think it deserves to be done."*

(Tiki Wiki CMS Groupware Community project administrator)

Therefore, the main role hobbyists play in the collaborative RDIP is to "maintain" the innovations that have already happened rather than realizing them or initiating them. But then again, maintenance of OSS is not necessarily inferior to doing "real R&D job" and innovation of the technologies in form of writing new modules and features to enhance the functionality of the software package. Because after all, as mentioned by the Tiki Wiki OS community leader, *"...if nobody is there to keep the maintenance of the thing [new features/innovations] and do the house-cleaning then there are bugs*

*all over the place then it doesn't matter if you have that fancy feature.*" The following quote highlight the role and significance of hobbyists /enthusiasts further.

*"In this case is not like live innovation but more of maintenance and clean-up! House cleaning! Because these are not the senior programmer, so they are not doing like big innovations but they are doing something which is necessary because you have your star programmers that do really the advanced stuff. But that's not enough! Because you can do all this fancy stuff but if nobody takes care of the documentation then nobody is going to use those features. And, if nobody is there to keep the maintenance of the thing and do the house-cleaning then there are bugs all over the place that it doesn't matter if you have that fancy feature. So, basically, these are also a fundamental pillar of the community. Because what they do is like drops in the bucket which in an aggregate form they also help sustainability and innovation of the community."*

*"in some cases because they have fresh eyes they find bugs because they are not going to write that core code but they are going to read it but it's mostly that they are going to solve hundreds of small little issues and if you add all that up...."*

(Tiki Wiki CMS Groupware Community project administrator)

### 5.3.2 Technological collaboration and its conceptual dimensions unpacked

"TECHNOLOGICAL COLLABORATION" is a multifaceted main category. Because of open coding and axial coding, I have managed to build this gestalt and abstract category through the emerging sub-categories of *context*, *sharing routines*, *relationships' typology*, and *dual-purpose leadership*.

#### A) The context for technological collaboration

First, technological collaboration within shared OSS RDIP is a context-driven phenomenon. This means that without having the contextual elements in place, firms cannot collaborate successfully. Based on data analysis, I group contextual elements together around the following concepts: *Know-why*; *know-who*; *know-how*; *Training*; and *Resources* (see Table 5.3).

For a firm to collaborate, it should first know ‘why’ this collaboration is necessary. The logic that underlines collaboration therefore refers to as *know-why*. In my interviews, all three actor groups (OSS firms, clients, hobbyists) and community leaders and developers unanimously agree that the fundamental reasons for collaboration are *project maintenance*, *sustainability*, and *inter-operability* of their version of software with the one that exists in public domain, the commons. *Know-how* constitutes another context-related concept and it refers to having the necessary technological capability, relationship building capability, and relationships through which a firm can connect and share knowledge. *Know-who* further refers to knowing the key actors in OSS communities to make a targeted interaction in order to make sure the resources are used both efficiently and effectively. An example will clarify these three dimensions of the context.

Odoo, an enterprise resource planning software is quite a popular and solid OSS project with a professional community around it. If one Canadian enterprise client or OSS firm intends to have a feature developed in collaboration with one of the experts in the Odoo community, the initiator needs to know whom to contact (*know-who*) and how to proceed with the R&D process (*know-how* or expertise). The experts in Odoo community are not all located in a particular geographic region, and certainly not all are in Canada. Therefore, in order to remotely induce a behavior in (i.e., to encourage participation of) an expert, there needs to be a channel (i.e., relationship)

between the collaborators. More importantly, this relationship cannot necessarily be a legally binding one as the individual expert (a software developer) who is in, for example, Belgium, cannot enter into a contract with a firm in Canada due to high transaction costs associated with concluding such agreement. After all, for a project that costs five to ten thousand dollars, it is often too costly to hire a lawyer and sue a foreign developer firm from Belgium; should things go wrong. Such relationship, therefore, are often built on trust and familiarity.

Table 5.3. Unpacking of the Concept of Context for Technological Collaboration

Context for collaboration	Definition(s)	Consequences
Know-why	<ul style="list-style-type: none"> <li>• Actor knows clearly about the underlying reasons for getting engaged in technological collaborations.</li> <li>• The underlying reasons are a mix of self-interest and group interest.</li> </ul>	<ul style="list-style-type: none"> <li>• Top management supports their IT departments' decision-making concerning resources allocation.</li> <li>• Adoption and integration of OSS becomes part of a long-term plan as opposed to a single-shot software acquisition policy.</li> </ul>
Know-how	<ul style="list-style-type: none"> <li>• Having in-house OSS technological capability such as OSS-specific absorptive capacity that enables an actor to get engaged in the collaborative software R&amp;D.</li> <li>• Having relationship-building capability so as to be able to connect with other collaborators.</li> </ul>	<ul style="list-style-type: none"> <li>• The user turns into an innovator because they are willing and able to collaborate.</li> <li>• In long-term, the internal know-how (expertise) boosts and ensures further on-going collaboration.</li> </ul>
Know-who	<ul style="list-style-type: none"> <li>• Knowing the influential and most capable people in target projects.</li> </ul>	<ul style="list-style-type: none"> <li>• The actor makes very targeted and effective investments in their relationship building efforts.</li> <li>• The actor increases the chances of getting the right responses through interactions, in a timely fashion.</li> </ul>

Source: Author.

On the plus side, know-how can be taught through training provided by OSS firms or it can be learnt through interaction with community members. Therefore, the firms can learn how to enhance their OSS-related absorptive capacities and relationship building capabilities through different stages of training. However, know-who is a matter of time and investments a firm makes in OSS communities and it cannot be built over night.

Furthermore, context for collaboration encompasses having *resources* and *allocating them appropriately*. Resources are; namely, having experienced OSS developers to assign them to projects, time to spend on collaborative developments with community members and get involved in iterative software development methodology. Resources are also a matter of 'know-why'. Almost all SMEs and even large firms suffer from lack of resources in their IT departments. Therefore, what makes them different in their resource allocation strategies is their perspective of why they should engage in this collaborative RDIP. Indeed, if a firm looks at OSS solutions as “a quick fix” with no long-term horizon, the IT department head cannot justify either spending of the status-quo resources or the request to obtain more resources.

## B) Code sharing routines and its complexities

Sharing is another sub-category that lies at heart of the collaboration concept. It involves code sharing with community of developers in public domain (i.e., commons). A key to successful OSS shared RDIP is *code, knowledge and resources sharing*. *Code sharing* does not only mean to build and develop technologies in form of features, modules; it also emphasizes the necessity to increase the *quality* of the written software and further make sure the project evolves in harmony with all its dependencies and branches. Therefore, it plays a key role in project *maintenance* and *sustainability*. Code sharing also ensures that all versions of the OSS remain



interoperable. Interoperability condition therefore acts like an invisible chain and binds all actors of the RDIP together as a unified system. The following quotes illustrate the benefits of the community members in maintaining code for the good of each member.

*"We cannot just say community is good and go for the community...We have to do it by ourselves [sharing back and contributing to community] otherwise the community dries"*

(OSS firm's developer-Linux kernel project)

*"The idea behind that [sharing, contributing back] is to enhance the quality of the tools (Linux, its dependencies, its ecosystem) we are using. We want to make them better. If we want them to be better, we have to contribute to communities.... Yes, also because we are not just using OSS, but also we are contributing to free software world. So, it's easier for us to get feedback from other project communities. We are not just like a big company using all the stuff and not giving back. If you don't give back, people would be less or a lot less willing to cooperate with you if you happen to just using, using, using and never sending anything back substantially."*

(OSS firm's developer-Linux kernel project)

*"That's what the OS is all about: sharing codes for the benefit of others. So we kinda try to really avoid scenarios where companies benefit from sharing a code but they use the community as a source to leach without giving anything back...."*

(BigBlueButton project leader)

Code sharing is also a very *delicate process* that needs to be managed carefully. On the one hand, it increases the cost of R&D and prolongs the development process due to iterations. On the other hand, it affects issues of *client confidentiality* and *security*. Therefore, managing the code sharing task and its process is a matter of knowing both the 'hard issues' (technical issues) and the 'soft issues' (client business-related issues) associated with its process. One key question is, "*What part of the written*

*code can be and/or should be shared with communities in public domain?” and “Who makes such key decisions— which may have (negative/positive) repercussions?”*

Based on the interviews, I can detect several important considerations associated with code sharing decision-making. First, the *type of licensing* can dictate the sharing task no matter what the preference of actors is. For example, if an OSS firm uses an OSS library that is under GPL licensing, the codes related to this component need to remain open source and must be shared with the community automatically. Second, OSS developers need to make a distinction between *useful and general codes* and non-useful, very specific codes. This is mainly because some written codes are so specific that may become irrelevant to the community of developers once shared, thereby remaining useless. This is in contrast with the case where an OSS developer comes up with a solid and improved packaging method that can be used by many developers in the community. Third, OSS developers need to distinguish between *secretive and non-secretive information* that can be revealed by chunks of codes where the code sharing exposes a client’s private business or technology secrets. These chunks of codes, however useful, must not be shared. Fourth, code sharing with the community means spending more time on R&D and adding more costs to the overall project. In fact, this iterative process involves many back-and-forth discussions with the community experts in order to get the code quality enhanced and arrive at a cleaned version of the code in accordance with community standards. The goal is to get the best possible solutions integrated into the core so that they evolve along with the remaining libraries and features as time passes by. Therefore, an OSS firm and client need to *negotiate the added time and cost of RDIP*, and share that cost proportionately. If a client does not agree with new deadlines or the estimated augmented costs, then the OSS firm will have to invest alone in this collaborative process. Fifth, code sharing reflects the *ethics* associated with OSS development process, which if ignored, it may have detrimental impact on the relationship of client

or OSS firm with OSS community. All of these are among key considerations related to code sharing process. The following quotes illustrate how managers balance the needs of community agents with those of their clients.

*"There is a gap between sensitive and non-sensitive information. Plus the experience you have in doing these parts. It counts really. For example, in monitoring software you do a lot of configurations. And you cannot really release them as these configurations contain IP addresses and sensitive information so you cannot release them back, for sure. So, it relates to the client's company so you can't release them. But your experience can be great for others too: the way you architecture your configuration and this can be great for the community. For example, I used a specific template configuration or module in my solution. I can give this hint to the community."*

(OSS firm's monitoring practice leader)

*"If we come up with something very technically specific and of no use to community, we will not share it back to community. So we build modules that are re-usable and we have site-specific modules for which there is no point to share that code at all."*

(OSS firm's developer)

*"On my first project it was exactly like that. The client wanted an application just for himself and was not interested in sharing the source code with the community. But I used different open source libraries and in the project we used two licenses. One license we used for the library and this was an open source license. So, it was for the community and the client couldn't keep it for himself. The rest of the project was exclusively for the client and he could do whatever he wants with the source code. It was in-house source code because it was all developed in SFL for the client. So the decision on sharing the source code and developed solutions can be partly made with the client present at the table of negotiation. It is the question of licensing. If you find a library that is useful for the client's project and it can be anything like custom components. So, we have to follow the original license of the library. If it is like GPL license, you can't refrain sharing. It must stay open source and we have to give back the code."*

(OSS firm's developer)

*“Our intention is not necessarily to contribute to OS. We do, I mean when we have to, like because of GPL and all these contracts. We do contribute back to community but this is not our intention...we are not doing what we do because we want to contribute. For us is a by-product [code sharing and contributing back to OS community]”*

(Enterprise client R&D manager)

*My focus was to clean a bit the code and push it back to community. If you want some code to be integrated into the Linux kernel, you have to make it the way they want. So the code has to be really clean and organized the way they want. You can develop code in many ways but not all the ways are clean and acceptable by Linux kernel.*

(OSS firm's developer)

*The process of code cleaning and integration is peer review process which involves revision, modification and submission of codes based on feedback provided by community in an iterative fashion. This kind of job is also called R&D project and part-time took me about a year to finally integrate the code...It took seven reviews to finally get the acceptance of the community and integrate the revised version. When we developed the enhanced version of the OS then we pushed it back to the FLOSS community....*

(OSS firm's developer)

*“Even though I find it hard to believe because you just have to erase the file (file license.texte) if you erase it you can remove its traces. You know it is the question of ethics. I also cannot keep it a secret because we are working for example on SFLphone together and if people see a huge chunk of codes coming from nowhere then it is not a good behavior. You have to give back credit to people who did it and say it is their work.”*

(SFLphone R&D personnel, developer)

Overall, the code sharing plays a pivotal role in collaboration process; having significantly positive consequences for the sustainability of collaboratively developed software technology and the community around it. Each actor groups merits from the code sharing process. However, as explained above, there are conditions that must be

met before code sharing. Table 5.4 presents the conditions necessary for code sharing as well as its consequences.

Table 5.4. Conditions and Consequences of the Code Sharing in OSSTC

Code sharing by	Condition needed	Consequences for OSS firms	Consequences for clients	Consequences for communities
OSS firms	<ul style="list-style-type: none"> <li>• Licensing type,</li> <li>• Client's agreement (permission),</li> <li>• Client's collaboration,</li> <li>• Code type,</li> <li>• Ethical considerations</li> </ul>	<ul style="list-style-type: none"> <li>• Gains reputation and popularity in OSS communities,</li> <li>• Builds trust-based relationships with community developers,</li> <li>• Signals its technological capabilities,</li> <li>• Attracts more clients,</li> <li>• Attracts potential future employees from communities.</li> </ul>	<ul style="list-style-type: none"> <li>• Reduces upgradability risks,</li> <li>• Increases supportability chances,</li> <li>• Ensures future compatibility of adopted software</li> <li>• Enhances the in-house OSS absorptive capacity (know-how),</li> <li>• Client's learning enhances and so does its internal R&amp;D experience (know-how),</li> <li>• Embarks on building trust-based relationships with communities,</li> <li>• Gains know-who</li> </ul>	<ul style="list-style-type: none"> <li>• Gains more R&amp;D resources</li> <li>• Access client's feedback and perspectives,</li> <li>• Strikes balance between technology push and market pull R&amp;D,</li> <li>• Has more chances for sustainability long-term</li> <li>• Increasing positive network externalities</li> <li>• Project maintainability increases.</li> </ul>
Enterprise clients	<ul style="list-style-type: none"> <li>• Willingness to issue permission,</li> <li>• Having context-based elements for collaboration,</li> <li>• Code type,</li> <li>• Licencing type,</li> <li>• Ethical considerations</li> </ul>	<ul style="list-style-type: none"> <li>• Approval of clients enables OSS firm to share code,</li> <li>• Support of clients adds to OSS firm capability to share code,</li> <li>• OSS firm can share added costs of collaborative R&amp;D,</li> <li>• OSS firm can invest more in its relationship with client and community</li> </ul>	<ul style="list-style-type: none"> <li>• Reduces upgradability risks,</li> <li>• Increases supportability chances,</li> <li>• Ensures future compatibility of adopted software</li> <li>• Enhances the in-house OSS absorptive capacity (know-how),</li> <li>• Client's learning enhances and so does its internal R&amp;D experience (know-how),</li> <li>• Embarks on building trust-based relationships with communities,</li> <li>• Gains know-who</li> </ul>	<ul style="list-style-type: none"> <li>• Same as above</li> </ul>

### C) Relationships' typology

Relationships are central to OSS development process. For example, an OSS firm like Savoir-faire Linux has forged a very strong tie with Odoo, Liferay, Debian, and Linux Kernel communities of developers and has officially developed strategic partnerships with the enterprises backing these communities. By doing so, the company has high level of resource-dependency; but, forging a strong relationship with the communities allow the OSS firm to mildly influence the technology trajectory within those communities and reflect or voice its clients' needs effectively. Based on the data analysis, new relationships emerge as categories and I have formed two major types of relationships; namely, *Dyadic*; and *Triadic relationships* (see Table 5.5).

*Dyadic* relationship refers to a relationship that exists between two actors; and they are of three types: 1) OSS firm and its client; 2) OSS firm and the community of developers around the project; and 3) enterprise client and community of developers. *Triadic* relationship, on the other hand, involves a tripartite relationship mode that exists among three actor groups during the collaboration.

The dynamic of dyadic relationships between an OSS firm and community of developers reflects knowledge sharing routines that are bidirectional and continuous. It also includes quite high resource dependencies. Power distribution is between OSS firm and the project community (mainly core developers and leadership). As there is no perfect power distribution, one of the two parties may sit in the steering seat and have more bargaining power in the collective decision making process.

Dyadic relationships between OSS firm and community of developers are mainly trust-based; meaning that, they are forged through a long term, and effortful

collaboration and relationship building work, mainly initiated by the OSS firm. They are the most fruitful relationships and the most costly ones as either of the parties has invested a lot of time, effort, and resources in order to maintain the relationship in a mutually beneficial state.

Table 5.5. Typology of the Relationships among Firms and Communities

Relationship type	Dimension	Nature	Associated issues	Related capability
Dyadic	OSS firm-project community	Mainly trust based	<ul style="list-style-type: none"> <li>• Power distribution</li> <li>• Resource dependency</li> <li>• Two-way Knowledge-sharing routines</li> </ul>	<ul style="list-style-type: none"> <li>• OSS Absorptive capacity</li> <li>• Relationship building capability</li> </ul>
	OSS firm-client	Contract-based		
	Client- project community	Contract and trust-based		
Triadic	OSS firm-client-project community	Mainly trust-based	<ul style="list-style-type: none"> <li>• Power distribution</li> <li>• Resource dependency</li> <li>• Tripartite Knowledge-sharing routines</li> </ul>	<ul style="list-style-type: none"> <li>• OSS Absorptive capacity</li> <li>• Relationship building capability</li> <li>• Leadership capability</li> </ul>

Source: Author.

Dyadic relationships between OSS firm and their clients are mainly *contractual*. Clients approach OSS firms with their IT needs and sign a contract that also includes a licensing agreement with the OSS firm. Therefore, there are clear objectives and expectations set from outset of the project. There is bidirectional knowledge flow between client and OSS firm in that the OSS firm learns about the BM and business affairs of the client organization in order to be able to customize most efficiently and



effectively the IT services and modules. The knowledge routine dictates client to contact OSS firm in case they need to highlight a problem, a bug or if they have any recommendations. Therefore, the client is not connected with the community of developers around the OSS project. The client's IT department also attempts to go through stages of training to learn about the fundamental aspects of the OSS solutions and tools being used in order to be able to use the technology and provide internal support if necessary. The power distribution is between OSS firm and client and is mainly based on the type of contract signed between them. If the client intends to influence the technology trajectory, they prefer to go through OSS firm because the client has no relationship with the on-line community.

Dyadic relationships between client and OSS project community members are mainly *trust-based* but they can also be based on formal contracts signed with the community leadership or governing body. As shown earlier in Figure 5.11, the 'OSS enterprise client typology', the types of clients that establish dyadic relationship with OSS communities fall under quadrants C and D. They normally have in-house IT personnel and expertise (OSS absorptive capacity) to connect with source of technology directly and share knowledge bi-directionally. There are some resources dependencies as clients directly rely on community resources concerning satisfying their IT needs. However, power distribution is not necessarily balanced and it depends on how significant client's role and resource allocation to project are as well as how strong the leadership and governance of OSS project is. According to interviewees, OSS projects aim to keep the "source of gravity" inside the community project by managing the influence of external forces on the project leadership. For example, there have been instances where the client's influence has been so high that they could even acquire the whole OSS project and turn it into a closed proprietary project.

In contrast, triadic relationships only exist when OSS firm, their client, and OSS community of developers engage in collaborative RDIP together by sharing knowledge, investing resources, and managing the process in a tripartite fashion. Building, managing and maintaining tripartite relationships seem to be simultaneously the most rewarding and the most challenging. It is the most rewarding mainly because all resources, knowledge and feedback, and technological capabilities merge together in the OSS development process so much so that the generated synergy as an outcome of the tripartite collaboration leads to (self-)sustainability of the development process. As client and community are involved tightly, the technology being developed responds almost impeccably to the client needs and at the same time draws on all possible available sources of development.

It is also the most challenging type of relationship mainly because the diversity/heterogeneity of participants is at its highest with each having *different interests, expectations, and level of technological capability*. While learning and experimenting together, they need to align their needs and wants, expectations, and remain steadfast on the innovation track in order to produce results and meet project milestones. In other words, they must form a goal-oriented network. Further, their continued technological collaboration is necessary to sustain the project over time and ensure long-term project maintainability. This in turn results in an OSS solution or tool that is highly interoperable among three actor groups. Therefore, they can continue building upon the collaboratively developed technology. Open governance and leadership are therefore central to holding the actors together.

Furthermore, irrespective of relationship type, actors must have two main capabilities in order to be able to collaborate and share code. The first capability is OSS absorptive capacity. Absorptive capacity originally refers to "*the ability of a firm to recognize the value of new, external information, assimilate it, and apply it to*

*commercial ends*” (Cohen & Levinthal, 1990, p. 128). In context of collaborative OSS R&D, it needs to be very specific as mainstream software engineers in corporate North America are mainly trained by an education system that highly favors proprietary software and is also promoted and supported by proprietary (software) corporations such as Microsoft, IBM, Apple, etc. Therefore, for either client or OSS firm to be successful in benefiting from OSS technology and be able to play a role in its development, it is highly critical to have in-house OSS expertise. This has also been a challenge for OSS firms whose main task is to offer services on OSS software tools and products. The following quotes illustrate the difference between OSS engineers and the mainstream ones as well as the shortage of OS engineers in the industry.

*“SFL does not grow very well right now. We do not have enough resources. Today I have to tell my customers that due to lack of human resources I cannot respond to their needs and cannot work on their projects. Market is big enough and we would like to hire more engineers but there are not good enough engineers out there. We are talking about open source software so we want engineer profile because they need not only know how to operate the system but also they need to understand how the system works. For example, a Windows technician does not know how the software works. He operates it, and that's all he knows. For every new version, he needs to be retrained; whereas in case of open source of course you need to know how to operate, but you need also to know how it works behind the scene.”*

(Odoo practice team leader)

*“There are no technicians in open source, there are only engineers. That's where companies gain productivity by having engineers who know how the systems can be configured to match the needs of the company.”*

(Odoo practice team leader)

*"Today it takes me about a year to hire and train someone in OpenERP department and bring him to speed with other team members. Recruiting and training a new OpenERP team member should be the job of middle manager."*

(Odoo practice team leader)

The second capability is "relationship building capability". It is especially important under circumstances where either client or OSS firm needs to directly connect with OSS project communities. Within OSS ecosystem, relationships are mostly trust-based, meaning that because individuals know each other or are bound with each other based on shared values and norms they engage in knowledge transfer and resources sharing. It is literally impossible to write and safeguard a contract for any activity a firm intends to do in public domain. However, there are big collaborative projects which may demand all parties to enter into some written agreements but overall it is a developer's relationships that get the job done. The following quote illustrates how communities provide support for OSS firms.

*"We have also this strong relationship with the community. We are like a gateway for anyone, for any company to reach the relevant community member. For example, last week the customer wants a feature that unfortunately I don't have the resources internally to provide it; and it would take a lot of time to explain the project and train one developer to be able to provide the feature. So, I just contacted a few friends from the community on the web, and they told me they could provide it in one day! So two days after, I got the module providing the feature. So I skipped the learning curve, I skipped the availability of my team and my resources but I am still able to provide the service through my relationship with the community."*

(Odoo practice team leader)

In addition to the aforementioned two capabilities, triadic relationship demands the actors to create some "leadership capability". As three actor groups need to share knowledge and resources together while having a heterogeneous set of interest and

expectations, the key to successfully identify, share and complete project tasks and meet the milestones is to have sound leadership capability. This capability is further discussed under a separate category as it encompasses a myriad of factors.

#### D) Dual-purpose leadership

The interviews with OSS core developers and individual hobbyist, those in project management positions in OSS firms and clients' organization and most importantly people in leadership positions in communities, have led to discovery of the DUAL-PURPOSE LEADERSHIP category. Indeed, if actors, collaboration and relationships form the *bricks* of collaborative OSS RDIP, the leadership is the *mortar* that glues all these constituents of OSS ecosystem together. More specifically, within OSS projects where it is a matter managing both *technology* (and all associated issues) as well as *people* through time, issues of management and leadership tend to merge together giving rise to the emergence of the key concept of "Dual-purpose Leadership". Table 5.6 demonstrates the breakdown of the concept.

Primarily, a dual-purpose leader of a successful OSS project is a visionary leader who knows or at least has a good sense of the destination of the project's technology. They tend to think long-term when it comes to the evolutionary path of the technology and they think of the "relevance" of the technology to the future's market needs. The following quotes illustrate how leadership means having foresight or the ability to discriminate the successful from the failing.

*"Leaders need to have vision; some idea with regards to the nature of technology, its evolutionary path and the future road ahead...So in terms of vision, I think, it helps if you have a global idea of where you want to go; so when things happen you can evaluate how they fit."*

(Tiki Wiki CMS Groupware Community project administrator)

*"Good vision and good leadership are important factors of a healthy community. How to know what you want and filter out irrelevant requests which do not fall in the scope of the project. A project that accepts everything from everyone is gonna be slow and bloated with too many features which are unusable, hard to fix and buggy."*

(BigBlueButton project administrator/leader)

*"And I tend to think long-term and you know think of what we have today which are really popular like Facebook and Apple. Today they are pretty big! Who knows in 10 years or 15 years.... remember the time when ICQ was a big thing and now is Skype!... So for me we have to be careful with stuff which is useful today because who knows if it's going to be around in the future...I want Tiki has the capacity to evolve as needs evolve! ...So the vision of Tiki is not to define something pre-determined, but to accommodate changing needs."*

(Tiki Wiki CMS Groupware Community project administrator)

*"The problem is if you don't plan for future when the future arrives we are screwed. So thinking long-term means sometimes okay let's do this properly is going to take more time today but in the future we will be fine."*

(Tiki Wiki CMS Groupware Community project administrator)

*"So my theory is our evolution path is way better than that. So because our evolution path is better we can evolve better and we can adapt to the circumstances of whatever technology throws our way."*

(Tiki Wiki CMS Groupware Community project administrator)

*"So whatever we do has to work in the next version as well so because we are always thinking of the future we are always keeping things future-proof."*

(Tiki Wiki CMS Groupware Community project administrator)

Table 5.6. Breakdown of the Concept of Dual-Purpose Leadership

Dual-purpose leadership	Building concepts
Vision	<ul style="list-style-type: none"> <li>• Having long-term vision,</li> <li>• Being future-oriented or forward-looking with regards to technology evolution path,</li> <li>• Planning for future and embracing the flexibility to adapt technology in the course of time.</li> </ul>
Setting objectives	<ul style="list-style-type: none"> <li>• Setting project's goals, objectives, milestones,</li> <li>• Clearly communicating project roadmap with community members.</li> </ul>
Managing technology	<ul style="list-style-type: none"> <li>• Potential for commercialization (closing the gap between clients and community),</li> <li>• Technological sustainability,</li> <li>• Economic sustainability.</li> </ul>
Managing the fear of forking	<ul style="list-style-type: none"> <li>• Leader's attitude (Reasoning with people; Favoring compromise over demise),</li> <li>• Request management considerations,</li> <li>• Technology management considerations,</li> <li>• Conflict resolution mechanisms,</li> <li>• Acting with resolution and determination in critical times.</li> </ul>
Managing/ leading the OSS community of actors	<ul style="list-style-type: none"> <li>• Managing the collective power to create synergy,</li> <li>• Coordination mechanisms,</li> <li>• Promoting concept of open governance, walking the talk,</li> <li>• Ensuring transparency in decision making process.</li> </ul>

Source: Author

Second, leaders tend to set clear goals, objectives and standards and communicate them with the members. They also organize the important milestones chronologically so that the community of developers works together towards agreed-upon deadlines and deliverables.

*"The goal is stability, usability, and features, in that order. So if the feature is not stable, it doesn't matter how useable the features are..."*

(BigBlueButton project administrator)

*"Have a predictable system, people know what to expect from us... to be able to have a solid commercial ecosystem. The consultants need to be able to tell customers what to expect! And IT departments need to be able to tell their users when the next release is coming out and when the new features are available. If you don't have that predictability that is just like nothing happens. Everybody is waiting for everybody."*

(Tiki Wiki CMS Groupware Community project administrator)

*"So if we have two options in form of systems which both function, we would go for the simpler one."*

(Tiki Wiki CMS Groupware Community project administrator)

Third, leadership of a collaborative RDIP places a great deal of emphasis on commercial viability and sustainability of the OSS project. The main idea is that if there is value-added R&D that solves a real market need, then the project will gain external contributions in form of human capital, testing for quality assurance (QA), and funding. These conditions collectively lead to sustainability of the project both in terms of technological as well as economical sustainability. As stated by the BigBlueButton project administrator (see the following quotes), a viable OSS project generate revenues, and creates benefits that can go back to the community.

*"I would argue that all of the successful Open Source Projects have very strong commercial interests where the commercial company sees the economic benefit of having a strong open source project."*

*"One of the factors in evaluating a community is to find out if it is backed up by an enterprise, or if the community is involved in offering a commercially viable or commercial solution. If they are R&D-contributed you can make profit out of it and therefore your OSS firm can invest more into that community. So, OSS firm takes part in enhancing the community and making it active. The more commercial companies that have embedded the open source software... Yes, you could say a code that is solving a market problem. You have a code base that is well structure and well maintained."*



*“An open source project with no revenue is not going to be viable in the long-run. A lot of times when you create an open source project from scratch, it's gonna be hard to sustain unless there is an ecosystem in which commercial companies can build upon and contribute back to the software...So what happens next is even if they ship and nothing else happens, the product has gotten more stable along those bugs fixed in the core, developers around the product has earned revenue, and you have a design win. You have reached a milestone in your open source project where a commercial company has embedded your software into their product.”*

(BigBlueButton project administrator)

In the absence of successful leadership and cohesion with the community, a project may lose leadership and eventually break up into its different projects— the phenomenon referred to by leaders as “forking”. Dealing with “Project Forking” is one of the most significant incidents that can happen (quite frequently) in life of an OSS project. Practitioners perceive this phenomenon as an indispensable part of the life of an OSS project. It can significantly influence the project’s future and the cohesion of the community around it. This phenomenon emerged frequently in my interviews, so I decided to design a few questions around it and ask interviewees more about it.

*“So eventually a fork eats up a lot of energy because a bunch of people start deciding where they are going to go because you may feel that this fork is initiated due to technical reasons so you are okay with it but then you are still losing some of your developers.”*

(Tiki Wiki CMS Groupware Community project administrator)

Overall, project forking is perceived as *a phenomenon that entails a split in the focus, labor and scope of the original community project*. It occurs when the community as a whole cannot reach an agreement over the direction of technology development. Thus, the developers part in two or more groups where each continues in a different direction and tries to evolve the technology differently. The disagreement can be due

to *interpersonal* conflicts, purely based on developing negative feeling towards one another, or *technological* conflicts or a *mix* of both. Many respondents tend to view forking as something negative rather than positive.

*Interpersonal* disagreement arises when two influential group of developers develop negative feelings towards one another which in turn result in lack of mutual understanding and cooperation working towards a common goal. Negative feelings can be rooted in “excessive prideful personalities”, “feelings of technological superiority” well beyond peers, ill-formed verbal behavior and manners, constant feelings of resentments, leadership’s hoarding the project’s authority and code base by being stubborn excluding others from being heard and taking a part in collective R&D process. Sometimes lack of delegating responsibility to other developers or lack of team spirit can lead to bottled-up grouses that later cause forking. Therefore, forks that happen due to interpersonal or soft issues are the ones that perceived as absolutely “*bad forks*” which are also mostly doomed to fail and shrive away.

*“The bad forks happen due to a political problem or even worse when there is a personal problem, like when two people hate each other; and, the reason for forking is because they can’t stand each other. That’s sad.”*

(Tiki Wiki CMS Groupware Community project administrator)

*Technological* disagreements on the future destination of the project seem to appear when a cluster of participants in a community proposes a specific technological trajectory that is fundamentally beyond the scope of the project. At least, the proposal leads to an intense argument. This is mainly the case where negotiations and discussions on finding the middle ground to keep different interests converging fail mainly due to unavailability of a middle ground regarding a hard technological solution. It is not necessarily due to soft issues (i.e., interpersonal disagreements). Although when forking happens people feel disappointed, the restructuring of the community based on technological grounds lead to survival of the main community

and birth of a new technology group (i.e., the fork). The forked project may or may not succeed in the long-term. Regardless of whatever the fate of the forked project may be, this is perceived a rather “good fork” compared to the previous scenario.

*“If you're creating the fork because you have reason to believe that that's the best thing for the project and thus you think forking is good. If you're trying to run the existing project and you are doing your best there and somebody does a fork, you may feel that you know they're just diluting the energies and we should all work together....”*

*“If a fork happens due to technical reasons, a technical disagreement on the future of the project and this is interesting because this is related to your question of when do you say no? .... I'm saying you always try to find a compromise. But sometimes you cannot find a compromise because the proposed technology is fundamentally different. And there is no middle ground because you cannot do both! Well then what do you do? So that is a good reason for forking if no one group can convince the other group.*

(Tiki Wiki CMS Groupware Community project administrator)

Given that forking is sometimes inevitable, *how do leaders manage the fear of forking or even resolve it?* Based on the data, dealing with forking phenomenon seems to be a matter of certain hard managerial skills and soft issues related to leading a group of people. First, leader's attitude is very important in approaching the conflictual situations to nip the issue in the bud. One project leader mentions that:

*“A bad compromise [that can be technologically inferior to the original plan] that brings people back to working together is more preferable than a [technologically] good decision which leaves people angry with bruises.”*

(Tiki Wiki CMS Groupware Community project administrator)

Therefore, a leader who intends to avoid conflict ‘promotes’ the attitude that accommodates different opinions through compromises or a middle ground solution.

*“If one group wants one thing then this group has to understand what other group wants. Everybody in the community has to understand what the other people need to be able to find a compromise. Is not necessarily the case that what I need is better than what you need. The case is that we all need stuff and we have a shared resource so the question is how we can work together to get all we need without causing problems and people are very reasonable so once they understand what other people needs are they are OK with it and then we find the solution. So there is a lot of negotiations there and this does not necessarily all go through the project leadership in a sense that if there are different points of views people will find a solution.... It is very rare that these negotiations go up to leadership level and we need to argue for that.”*

(Tiki Wiki CMS Groupware Community project administrator)

Furthermore, leadership is aware that conflicts are mainly due to diversity among community developers. Heterogeneity of developers (actors) lead to various needs and wants. Therefore, they tend to manage the diversity and resulting conflicts through certain initiatives that I group them under ‘request management considerations’:

- Distinguishing between needs and wants,
- Respecting innovators’ needs and wants,
- Trying to understand and analyze the requests from a technology development perspective and how they will impact the whole project such as what are their long-term effect on project sustainability, how do they fit with already existing features and project roadmap, how can these new requests (if responded to) be supported and maintained long-term,
- Trying to go for experimentation rather than straight cold reject as much as possible,
- Conducting post-experimentation evaluation to reach a consensus,
- Making a collective decision.

Thus, it is quite clear that request management considerations are part of leadership's attitude towards reaching a feasible compromise and taking a middle road rather than upsetting the collaborators up front. The key is to remain as a team and function harmoniously as one. The leader's attitude, the way s/he handles diversity and how s/he orchestrates the team effort towards reaching milestones play a critical role in managing conflicts.

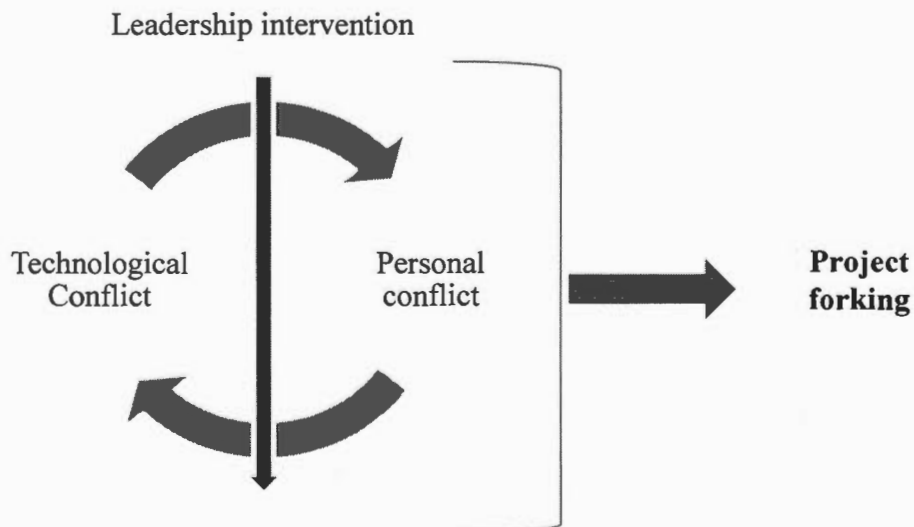
However, leading a community is not all about soft issues and managing people's requests. It can also involve hard issues associated with technology management. Indeed, compromises can happen at technological level too. This study's data show that leaders can break the vicious circle of conflicts leading to community break-up (see Figure 5.12) by resorting to certain technology management-related considerations (i.e., the hard side of dual-purpose leadership) so as to keep technological disagreements at minimum. Interviewees report the following methods.

**1) To avoid building an over-complex system with large breath of scope:**

Over-complex systems end up having a large number of technological features targeting very different market segments. This means the project becomes a cumbersome one having several centers of gravity beyond the core parameter (e.g., Drupal). The project therefore is hard to manage. For example, as many extensions are heavily developed by one segment (e.g., different firms' IT departments); managing all segments cohesively becomes a serious challenge for the community leadership. Therefore, many extensions may become obsolete, some die and some others evolve into a direction where they cannot become integrated back into the core. Some of forking based on technological conflicts reflects these cases.

- 2) **To consider skill and competency level of community contributors:** Some communities enjoy having a wide array of developers with varying levels of skills and competencies. There are senior programmers who can handle complex coding systems; average programmers who shy away from too complex systems; junior programmers who can only work with parsimonious systems; and power users who are sporadic code writers with limited capabilities. Once the leadership cannot coordinate properly these differences, one group may start a fork based on lack of technical compatibility.
- 3) **To know about technology positioning and act accordingly:** A community project is similar to projects in free market economy in that you need to offer a technology which has certain attributes and serves a range of purposes. This helps keeping the developers concentrated on the project goal and come up with innovations that are fundamentally aligned with project goals. In cases where there is opportunity for a technological forking, the two projects do not cannibalize on one another's target segment as each addresses a different set of technological issues.

Figure 5.12. The Vicious Circle of Conflicts Leading to Project Forking



Next, leadership uses “*conflict resolution mechanisms*” to organize the community of developers (OSS firms, clients, even hobbyists), as well as the decision making process; and, it provides a channel for resolving conflicts. These mechanisms may include monthly webinar, informal get-togethers, and casual face-to-face meetings over a beer or coffee just to discuss the itching matters. These informal mechanisms capture the very social and human aspect of conflict resolution, and they can be even embedded in formal coordination mechanisms such as yearly conferences or happen independently. The key to reducing tensions and resolving conflicts is to have an “*open, direct and friendly communication*” with different parties and to facilitate constructive discussions. Conflicts can be largely avoided when people have already developed a social relationship and the intended resolution emerges when people know each other both on personal as well as professional levels. The following quotes clarify this issue further.

*“Just by speaking and drinking some beer. No I am very serious! We do face-to-face meeting at each Eclipse call or con and drinking some beer perhaps is a better way to do collaborations with different companies in open source projects. I am very serious about that. And after that you know this person you are in contact in the mailing list, you know his taste; you know he is smart, and you don't want to.... okay, let's discuss, let's find a way to do something which is good for you and me; and that's just because of you! We are humans, we are not robots. If you would like to work with somebody else you would work better and it will work better when you don't care about the strategy of your company but when you know each other and you appreciate each other's perspectives and find a solution in the middle.”*

(Eclipse foundation board member)

*“And one of the things that we've done a lot is a lot of physical meetings called Tiki Fest so we travel we go to cities....Nowadays, most of the active people have met quite a few of the other active people so even if there are some disagreements it is difficult to tell someone to \*\*\*\* off on the mailing list if you had a beer with them last month.”*

*“So if you want to avoid forking and have a healthy community in general you need to have a conflict resolution mechanism. So this has a lot to do with better communications so for example one of the things that we have is that we have a monthly webinar.”*

*“So of course we have discussion forums and mailing list where people can ask questions and anybody can make a proposal but somebody could make a proposal that falls between the cracks and he may feel that he did not get an answer. But in the monthly webinars which last two hours in monthly community conference call well a lot of the main people are there if someone brings up the question they are gonna get an answer. So there is at least an opportunity to talk about issues. Just to be clear, if you want a bug fixed that does not mean it is going to be fixed. It is a do-acracy (when you do something, then you decide). Nobody works for anybody, but people will try to help each other and people appreciate that. So I think the fact there's a lot of discussions like that it reduces the tensions.”*

(Tiki Wiki CMS Groupware Community project administrator)

Finally yet importantly, leaders tend to reserve the right to intervene on occasions where their determination and prompt action can stop a damaging behavior initiated by some people – also referred to as “*poisonous*”. Poisonous people are described as those who are problematic by their very own nature; meaning that, they abuse the openness associated with commons in order to disrupt the collective flow. The following quote describes them from a leader’s perspective.

*“When you have an open community it is really cool and that works well but every once in a while we get some negative people called as **poisonous people** in Google or **toxic people**. And, this is a typical thing in an open community. These poisonous people typically have issues and they use the openness to poison the project and they are always negative and they always find problems.”*

(Tiki Wiki CMS Groupware Community project administrator)



So on these occasions, it is necessary for the leader to step in, in a resolute manner, and dismiss those selected individuals who just tend to disturb the flow no matter what.

*“You just let people do stuff but when there are poisonous people it's important for the leadership to deal with them. So we have not had them for several years but we have had before and at some point you have to tell them you have to GO! And that's difficult for an open project because in an open project you always try to get people so you don't have the reflex of kicking someone out! But sometimes you need to do it and that's a power that the leadership must maintain, that's the capacity to expel someone!”*

*“At the end, there is always a handful people in leadership position. That is the nature of things and the reason why these people are in leadership position is that other people trust them. And they think, okay, well, if that's what they think we will move along with it until another group of people takes over and replaces them. So we have to protect the community; and, if there's some people that are difficult then we warn them and eventually expel them from the project.”*

(Tiki Wiki CMS Groupware Community project administrator)

Fifth, dual-purpose leadership is all about “*managing and leading the OSS community*” and the diversity embedded therein. The power of the OSS communities lies in their knowledge diversity and the synergy they can create. Based on interviewees, I can define synergy as “*how a mutually synergistic combination of distinct capabilities, knowledge, and resources becomes more than their individual sums*”. By this, I mean that the advantageous combination of knowledge, capabilities and resources of the three actor groups (OSS firms, enterprise clients and hobbyists) lead to an added-value which cannot be achieved through combination of any of the two such as two OSS firms, or one OSS firm and one client only. Nevertheless, to manage and lead open communities (i.e., ambidexterity built into dual-purpose

leadership), leadership tends to mind several issues such as *coordination mechanisms*, *open governance*, and *transparency in decision-making process*.

Coordination mechanisms play a pivotal role mainly because the landscape of OSS development and diffusion has been changing, so much so that strategic OSS projects are evolving into a different form. Let me explain this issue further.

It is true that even today most of the OSS projects are created based on an individual's itching need, passion, and/or curiosity. In a sense, one person invents the technology and attempts to develop a community around it. For example, consider "Tig" and "LTTng" projects' beginning years. These are but just two instances I have studied by way of interviewing their core developers and community leaders. I categorize these projects under the category of "*Individual-based OSS development and diffusion*". The following quote clarifies how these projects started by an itching need of an individual.

*So the founder started few of these project in an experimental phase during his PhD. Mathew is still a Linux Kernel developer and he is widely recognized in the Linux Community as "the guy for tracing!" So he is really an expert in that field and another field which is RCU. So Matthew started contributing to Kernel during his undergraduate study. So he tried to find out about how do I have issues with my systems? How do I debug them at kernel level? At the time there was no easy solution. So he started the project of tracing the kernel to find out what is happening exactly and to be able to debug the specific system or problem. So it was technology push in a sense that he was scratching this issue...maybe as there was some research interest in that area and also there was his own personal need as he was working tightly with the Kernel. So there was research interest and personal need.*

(LTTng Project R&D)

There are also other categories that I have developed concerning OSS development and diffusion trend. The second category: "*OSS firm-based OSS development and*

*diffusion*” class includes those OSS projects which are created and initiated by OSS firms. For example, Savoir-faire Linux initiated two projects (SFLphone or Ring, and SFL vault) and turned them into open source. These are part of the in-house R&D strategic initiatives where management and leadership are centralized within the OSS firm. Thus, the center of gravity of the community lies within the firm. Figure 5.13 illustrates the typology of OSS projects development and diffusion developed in this study.

Although these two categories lead to formation of OSS community of developers, since their “*center of gravity*” (management, leadership, key decision making) is firmly located within a small group of individuals (one or two key developers in case of the first category) or affected significantly by the firm policies (in case of second category), the dynamics and diversity of community may be limited. The following quote from LTTng community illustrates this lack of diversity.

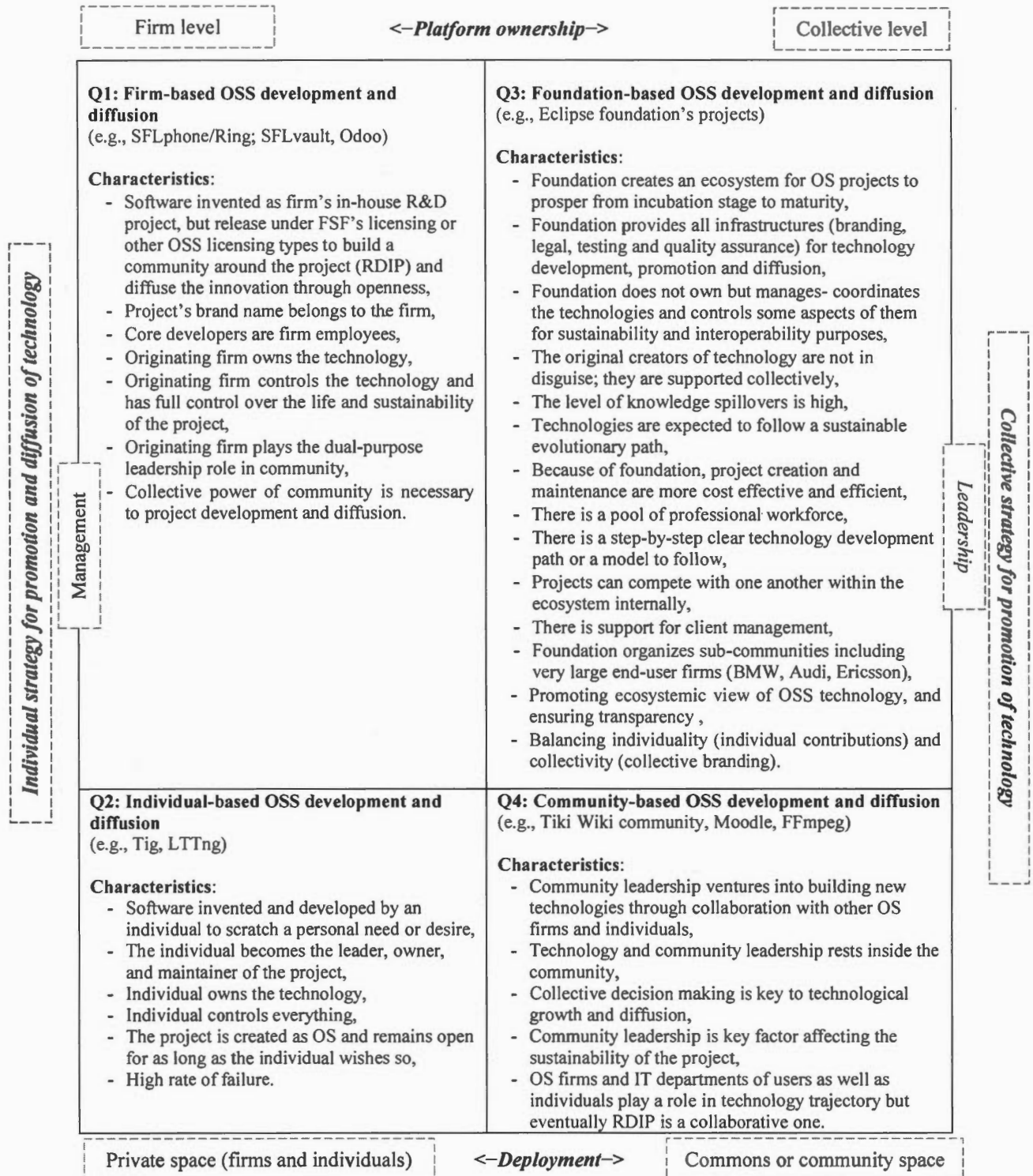
*“So one of the projects that Matthew is an expert on it is quite research oriented. Is leading edge on the technology side. And Matthew is one of the few experts in the world on the front...When the center of gravity is within the project that brings in interesting challenges! Like how do you ensure having a healthy community? If all the expertise lies within one guy, and that one guy gets hit by bus ....I mean bus factor! ... Matthew is maintaining 3 projects and the other 2 guys maintain each one project. So we have 5 projects and I contribute to all. There are other people who contribute but maintainer's job is really a different thing. Maintainer is like a leader. So they decide on which patch is ok to enter and which patch is not ok to be integrated. Or even determining the direction of technology is to a certain extent dependent on the maintainer.”*

(LTTng Project R&D)

The third and the fourth categories of OSS projects are the ones that are closer to the concept of strategic OSS I have been pursuing in this research. The third category is “*Community-based OSS development and diffusion*” like Tiki Wiki community where

the OSS project is managed and led by a group of key maintainers and the community around the project is very diverse and dynamic. The fourth category includes “*OS foundation-based OSS development and diffusion*” like Eclipse foundation which hosts over 250 OSS projects. Issues of *coordination mechanisms*, *open governance*, and *transparency* in decision-making are among the key factors that ensure that the community is managed and led towards generating synergy. Therefore, these factors have become all critically important in these two types of OSS development typology.

Figure 5.13. OSS Technology Development and Diffusion Typology and Strategies



Coordination mechanisms are important tools in hands of dual-purpose leadership to organize: a) the diversity of community of developers (OSS firms, clients, even hobbyists); b) the decision making process; and c) to create the synergy. Coordination mechanisms also act as *indirect conflict resolution mechanisms* to prevent or stop disagreements from blowing out of proportion and disturbing the synergy.

*Coordination mechanisms* include creation of associations or foundation – depending on the policies of the country in which they are registered. These associations help the leadership to manage different projects and diversity of contributors. In these associations, like formal organizations, different positions are created and clear roles are assigned. By breaking down responsibilities and tasks through delegating authority, decision making can be decentralized and therefore individuals can act more effectively. As a result, people can be more easily heard, objectives can be set and communicated collectively, and feedbacks are gathered and taken into account duly. Scheduled formal conferences and seminars as well as newsletters are among the tools which ensure people that there are formal scheduled channels to just get together and communicate. Furthermore, issues like scheduled releases or testing and quality assurance are managed through these formal coordination mechanisms.

*“Inside Eclipse several hundred companies work together and all the companies release on the same day...There are just some rules in place and we do the walk together on the same timeline. It takes a lot of coordination. For example, if you want to be on the release train, all your tests must be green. Testing and quality assurance must be passed. Before April there is an IP freeze. You cannot change your IPI anymore. Eclipse automatically uses tools to ensure that you are not changing your IPI. And if you make any changes, you can keep continuing but you are not any more on the release train. And this is bad for the visibility of your project. So these kinds of rules force you to be better and enhance your quality.”*

(Eclipse foundation board member)

*“Architecture council sets the standards. They are ten strategic members and persons who are elected by committers...Now the architecture’s work is to say or suggest what the proper ways to do plug-ins are so that they all work together...Voting is coming from “all” committers, non-members too. 1 or 2 persons are elected by members of Eclipse.”*

(Eclipse foundation board member)

*“So most of the active members of the community got together in France, Strasberg and we had a meeting for several days and we tried to decide about what do we do? We all agreed that the current system was bad. OK! So we came to the conclusion that the best way to do it is to have fixed release schedules, every 6 months there's a new version but every 3rd release is called the long-term support (LTS).”*

(Tiki Wiki CMS Groupware Community project administrator)

The case of foundations illustrates how organizations create and enforce coordination mechanisms. Eclipse Foundation, for example, performs: a) project management related tasks; b) managing legal issues; c) managing branding and marketing issues.

Project management tasks include establishing procedures for: project initiation, on-going monitoring the projects’ activity level, evolutionary path, quality assurance, and conducting maturity assessment. (See Figure 5.14)

Eclipse foundation ensures that there is a clear path for OSS developers to initiate and launch projects by creating standards and putting in place a healthy ecosystem where there is collaboration and competition so as ensure the projects are active and relevant to the needs in the markets. Monitoring of activity level of the projects ensures that the foundation is not hosting dead or dormant projects. Moreover, the foundation provides engineering architecture advice and services for quality assurance checks to make sure all projects are compatible with one another technologically and evolve as a family of interrelated technologies. The foundation also looks into maturity level of

projects to ensure they are progressing through different stages of development and are not stuck in their development process. The following explanatory quotes further clarify the tasks of a foundation.<sup>72</sup>

*"We can create a feature with better architecture and easier maintenance as the feature is now inside the core project. So a large part (more than 30%) of our business concerns evolution and creating improvements inside the OSS projects for clients."*

*"If a project does not have any activity during 2 years, the Eclipse project will approach the project owners and ask if they want to evolve the project. If no one cares anymore, the project will be killed and archived. So each project in Eclipse is alive! If you go to [sourceforge.net](http://sourceforge.net) for example, there are many projects that are not active. You don't know if they are alive or maintained."*

*"Or for example, we may have a customer who raises awareness over a ticket and show it to us, but then we say, this is not a problem on our side. It only concerns your technology. It may be a problem of compatibility with a product we just don't care about. Then we would say fix it yourself. They may exist and say you have to fix it because it is your project, but we would say no we don't care. So if the client exists we would say, pay the contract, pay the support and we fix it for you and for your own usage. And also 30% of turnover comes from bug fixing and making this kind of improvements. If you want a new icon to do a new action inside the project, then yes, good idea and we will do it in 2016! But the potential client, the user may say no I want it in 2 months from now, and we would say, ok then pay for it. And it is good for everybody because usually customers when they use proprietary software, if they want to add a new feature or request for it and pay for its development, they would say no. They would say, it is our own road map and we don't want you to pay us to do something which is not on our road map."*

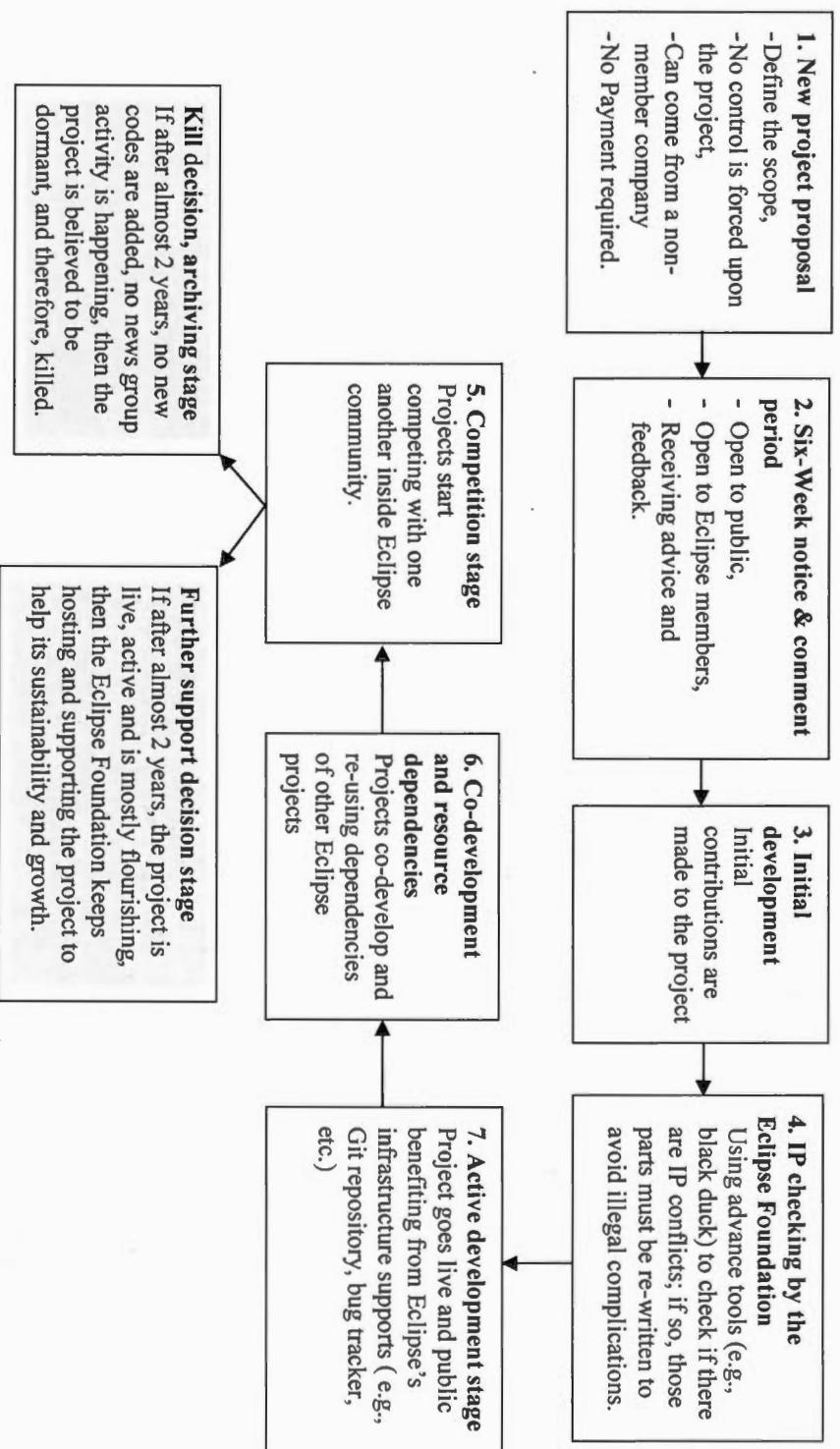
---

<sup>72</sup> Maturity assessment is quite a new concept at Eclipse Foundation. For them, it is an on-going issue to tackle. To conduct maturity assessment, they rely on different data from sources like: the activity on news group; the re-activity of the news group; the number of tickets; which tickets have been improved; which ones are still buggy; how long does it take for a project to fix critical bugs; how many users a project has; etc. Together, these metrics provide a notation on the question, if a project is mature or not. It is not only about quality, but it is about maturity. So to determine project maturity is quite a complex topic.



*“There are some services for quality checks. First, Eclipse provides some architecture advice. So inside Eclipse there are 2 kinds of projects. Projects that are part of a release train and those that are not part of a release train. Once a year in June on the same day we release about 100 projects. All these projects are compatible with one another.... Inside Eclipse several hundred companies work together and all the companies release on the same day. But in Eclipse, it is not on top of other projects. There are just some rules in place and we do the walk together on the same timeline. It takes a lot of coordination.”*

Figure 5.14. Seven Stages of Development from Project Proposal to Proof of Concept and Beyond: The Role and Function of the Eclipse Foundation



Source: Based on interviews with Eclipse Foundation

The Eclipse Foundation is also responsible for taking care of the legal issues. Since commercialization of strategic OSS project has come to the fore on agendas, the foundation needs to be sure that professional enterprise clients will not shy away from incorporating OSS projects into their products and services on the basis of complications associated with legal issues.

*“Eclipse only controls legal issues and requests project creators to be specific about the scope of their projects. Eclipse has human capacity and technology (tools) to check codes and avoid duplication on somebody else’s copyrights. The foundation is very meticulous about that to ensure that no legal issues will arise later. Each line must be checked and controlled.”*

*“The first reason is that Eclipse has an IP check (Legal Point of View). When you are a software provider like Obeo and you target large professional corporations, it is very important for them to make sure that upon using the technology they are not faced with any legal issues about it. When you re-use a code from GitHub it is possible that part of the inside project code is protected by GPL licensing. Just imagine you are Airbus and your software has a GPL protected embedded library. Then, all the code of the aircraft becomes GPL. Can you imagine that, that’s too risky. With this kind of foundation you are sure that the software is free from IP issues.”*

*“With this kind of foundation, even the process of IP control within foundation can be controlled. So commercial corporations can force the foundation to use the same process they use internally. For example, When Oracle needs a new tool for cloud, they say that you can use any project of Eclipse foundation without being forced to go through the internal process of IP checking at Oracle because we know that the Eclipse Foundation do the job by themselves. So for companies like Oracle, that is the only way to use OSS.”*

Finally, Eclipse Foundation manages the branding (logo, trademark) and marketing issues in order to lead the projects towards having a strong brand equity and worldwide recognition. They provide a “channel to promote the project” on their

webpage so that the newly initiated and run projects can benefit from a well-known and highly visited platform. They also provide chances for project leaders to become a speaker at Eclipse conferences in order to promote their projects within the relevant community.

*“Eclipse foundation provides the trademark to broadcast at a worldwide scope. In GitHub, you just post your project and then everything else is on you. No one knows your project; while inside Eclipse you have a channel to promote it. Eclipse webpage is a good place to promote as it is famous. Then there is Eclipse Con (Eclipse Conference) where you have a chance to become a speaker and promote your project. Sourceforge.net or GitHub provides hosting for the code but they don’t provide any infrastructure for quality check, long term support, etc.”*

*“Or, the other way, you create a technology through a network of companies under a foundation brand name, like Eclipse Foundation. Something that is original inside Eclipse—what makes Eclipse different from Apache Foundation (exactly the opposite)—for example, is that Eclipse allows for an ecosystem to exist. The foundation provides the logo, trademark and all infrastructure for technology promotion and diffusion. This leads to community-owned technology. In ecosystem, it is very easy to see who creates the technology. It is easy for a project to find training, professional services, support from providers. In other foundations like Apache, they want to hide the technology creators. For example, Apache foundation provides some tools for enterprise management like SAP. They don’t want to create an ecosystem. For example each email of a committer in Apache is @apache.org. So you do not get to know who is behind the technology (as an organization). You know name of the guy but not the company he is affiliated with. In Eclipse is the opposite (@oboe.fr; or, @ericsson.com). We want to do business funding where you can do promotions. Because we want to create leadership not only on the technologies, but also from a company point of view.”*

Furthermore, the dual-purpose leadership model ensures *open governance* is in place. A CEO/founder of an OSS firm – which is tightly involved with LibreOffice community of developers and particularly the core development team – emphasizes

the difference between viewing OSS from an open licensing point of view and open governance model. He notes that:

*“Samsung for example instead of building a real open source platform, they just take OSS technologies, hire engineers and develop everything in-house with no open governance model. It is open source by license but NOT by governance model! So that’s the main difference, right. If you want to take full advantage of the open source technology and open source methodology with everything it has to offer it is not about the license. The license is just a bit of a primary tool, the real advantage of open source lies in the real open source development methodology and an open governance model. And this is where you will get the long-term benefits of innovation and costs reductions. It’s always about cost. If you install a proper open source system and build a real open source platform, you will really have medium to long-term cost and innovation effect. And the problem is big companies like Samsung they don’t look at the long-term benefits, and not even medium term. Their entire focus is the next quarter. What is our next quarter look like? Companies still don’t know how to engage in open source communities. They want to control everything. Controlling and short term thinking are two main hurdles of moving towards open source. The companies like control and short term thinking.”*

(Collabora CEO/Entrepreneur)

As earlier discussed, the nature of dual-purpose leadership requires ambidexterity to lead and manage OSS collaborative RDIP. Naturally, there is need for putting a visionary leader in place. It is about promoting *real OSS methodology* that can only survive under real “open governance model”. Open governance is about *meritocracy* and *delegating authority and power* to influence the direction of technology trajectory to those who do most and make most significant contributions.

*“The key word in open source is governance model: basically how this project is technically developed? Who makes the decisions and who develops and contributes? What are contributors allowed to do and what they are not allowed to do? Each project has its own governance rules. All based on the same criteria that anybody can contribute and it is meritocracy so the more*

*you contribute the more influential you become and the more rights and power you have within that project.”*

(Collabora CEO/Entrepreneur)

And “transparency” is an aiding tool, an enabler that is necessary condition for establishing open governance. It permits actors to study contributions, evaluate them and probe further if necessary so there is no secret garden to hide from others.

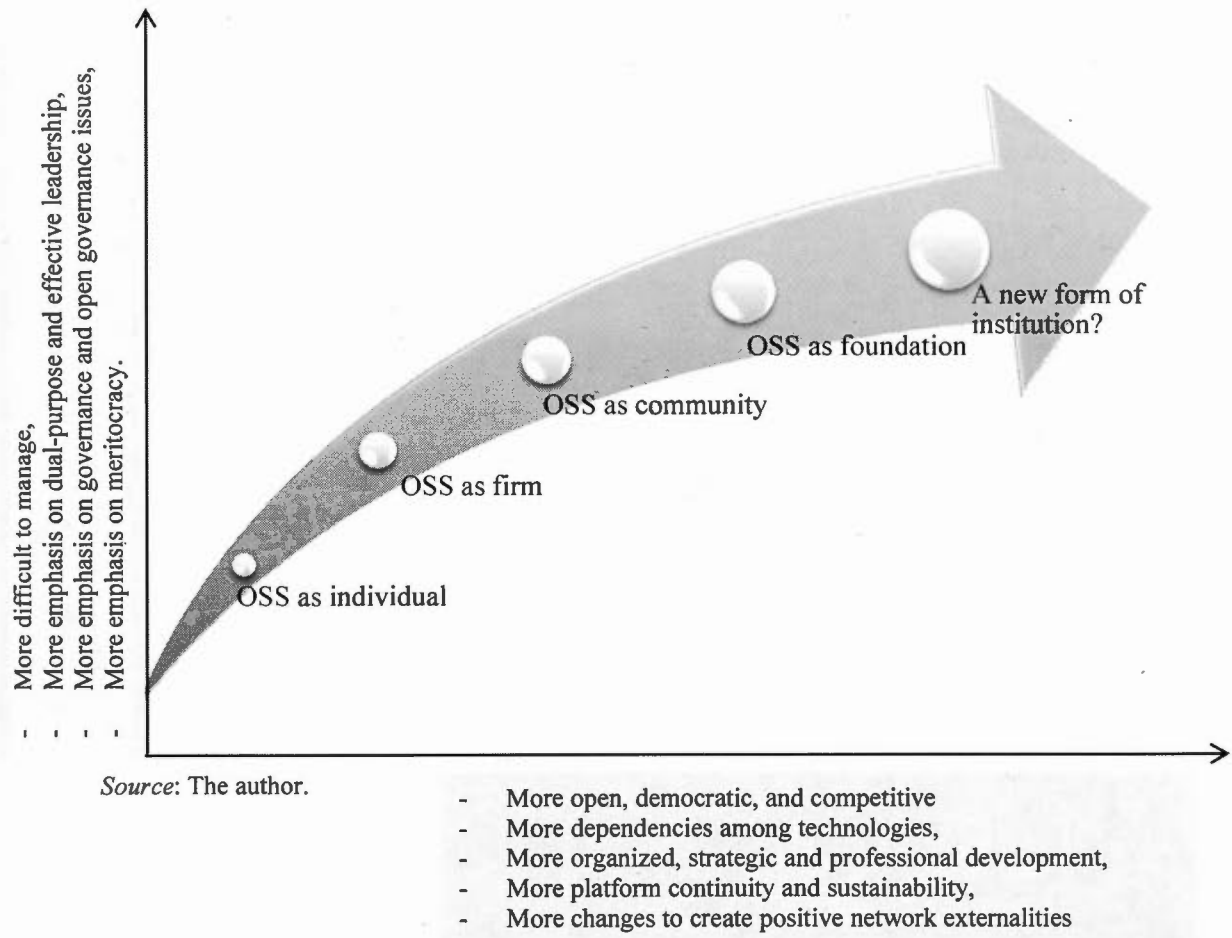
*“You can study and evaluate the contributions of the committer as a person and the company he is affiliated with. So when we say Obeo today is the 4th biggest contributor in Eclipse community; we can prove it as it is public and data are available public.”*

(Member of Eclipse board of directors)

Overall, my data point toward a new trend that highly emphasizes creating foundations or associations of OSS projects. In other words, this new trend is necessary to develop professional OSS RDIP that in real world can respond to professional and strategic needs of corporations (enterprise clients) and government bodies timely. OSS project leaders seem to be more inclined than ever before to forge borderless clusters through foundations and associations in order to orchestrate via dual-purpose leadership (manage and lead) the existing islands of OSS projects towards effectively meeting market demands. The dual-purpose leadership of a foundation also further ensures maintainability, interoperability and sustainability of projects. Figure 5.15 depicts this trend schematically.

As Figure 5.15 illustrates, the OSSRDIP is evolving towards Open Technology Platforms (OTP), emphasizing the “power of platforms” (Parker et al., 2016). OSS technology platforms, however different from the mainstream understanding of platforms as represented by Uber, Amazon, YouTube, etc. (see Parker et al., 2016), they share similarities such as being able to create network effects, and benefit from economies of scale and scope.

Figure 5.15. Institutional Forms and Institutional Evolution of Open Source Software



### 5.3.3 Interdependence-Independence

Having developed and explained the main categories of actors and collaboration along with their subcategories and conceptual building blocks, I now introduce the conceptual category of “INTERDEPENDENCE-INDEPENDENCE” as one of the main categories of this study. The concept of INDEPENDENCE refers to the fact that each actor is an independent entity in the OSS ecosystem. An OSS firm, an enterprise

client, or an individual hobbyist each has their own agenda to engage in OSS technological collaborations. They are also free to choose how to connect with each other and share their knowledge resources. For example, a client can very well choose to only deal with their OSS firm service provider through building and nurturing a dyadic relationship. Another client, however, may prefer to initiate and continue the collaborative R&D project from the very beginning through the triadic relationship encompassing its IT department, an OSS firm and the community of developers around the OSS project. Still another option for a client is to simply download the OSS tool and continue its development through its own closed IT/R&D department.

Furthermore, the *level* or intensity or involvement in R&D project as measured through amount of resources devoted to prototyping and full feature development, or amount of determination to create necessary OSS absorptive capacity and technological capabilities through long-term investments, is a matter of choice at the discretion of individual actors.

However, the notion of independence does not nullify, opposes or minimize the importance of the concept of INTERDEPENDENCE. Rather, the two notions are interwoven. Within the collaborative OSS RDIP, each actor plays a unique and specific set of roles which is complementary to those of the others.

As a case in point, consider the role a client enterprise can play in the commercialization process of OSS technology. OSS projects that are initiated by hackers and geeks (i.e., enthusiasts) in the OS communities may or may not have any potential for commercialization. When Matthew—the master mind behind LTTng project— first tried to submit his proposal in form of a set of codes to Linux community, his proposal was not received with open arms, nor enjoyed much



community support. It simply did not work “because community was not interested”. Luckily, he kept working on his technological ideas out of passion through his PhD work. Neither the community nor the inventor himself knew about the commercialization potential of this Linux kernel tracing tool technology. Only later on, when Matthew finished his PhD research, he became serious about career building and commercializing his project. More precisely, Matthew’s colleague said:

*“He tried at first to submit the code to the community but this did not work! Because community was not interested. But at the time Matthew moved on his own he did not realize there was a market for the project to build a company for that. That was not his first goal. His goal was to provide the tool to the community anyway possible. That was very altruistic! That's a lot work... that's the open source way! You provide value before asking for commercial side of it. So he did that project during his PhD study. So, after that he wanted to continue his PhD work, he taught, so why not start my company and be my own employee. And then some people found the project cool and useful and expressed their needs. So we provided consulting. And that's how EfficiOS got started.”*

(LTTng project R&D)

Had there been other OSS firms or enterprise clients involved in the process of decision making over Matthew’s technological proposal, things could have been probably turned out differently. Fortunately, LTTng became commercialized through the entrepreneurial effort of its inventor leading to inception of the small enterprise called EfficiOS. The community around LTTng also grew bigger and gained popularity so much so that LTTng was deemed as one of the accepted OSS projects to host a mentee for code writing mentorship program<sup>73</sup> called “Google Summer of

---

<sup>73</sup>The project abstract: Recently, with the rapid development of cloud computing there is a huge movement towards distributed data aggregation and Big Data analytics. Although there are many tools that deal with the aggregation and visualization part, like Facebook’s Scribe or Twitter’s Zipkin there are few tools that deal with trace creation and how tracing can have the least possible overhead. I am interested in connecting LTTng’s low-impact nature with the scale and visualization capabilities of Scribe and Zipkin.

Code”<sup>74</sup> in 2014. The case of LTTng project highlights the role of *diversity of actors* and specifically inclusion of perspectives of industry clients and OSS firms that tightly work with them in detecting commercial value of an OSS project.

Perhaps the concept of *triadic relationships* (as a subcategory) discussed under technological collaboration category can also be viewed as yet another example which leads us to form the notion of *interdependence* among actors, roles, resources, and relationships. In fact, interdependence is a meaningful concept because there are independent actors in the ecosystem. A firm, an organization or an individual needs to be independent in order to be able to benefit from interdependence.

In the process of interviewing the IT department head (also playing role in strategic management of the firm) of a large enterprise client, I realized how rewarding and crucial the triadic relationships could be. In this particular instance, the enterprise client makes a conscious decision through strategic IT-related discussions with the OSS firm’s consultants to forge a long-term strategic tripartite R&D relationship with an OSS project called ‘Shinken’. The tripartite relationship was forged through connecting a development team from client’s IT department with one of the leading core developers of Shinken project and additional expert support from OSS firm service provider. The idea was to avoid acquiring the OSS project through directly buying the whole project and turning it into closed software. Instead, the client

---

(Source: <https://www.google-melange.com/gsoc/project/details/google/gsoc2014/marioskogias/5668600916475904>).

<sup>74</sup> “*Google Summer of Code*” is a global program that offers student developers stipends to write code for various OSS projects. “We work with many open source, free software, and technology-related groups to identify and fund projects over a three month period. Since its inception in 2005, the program has brought together over 7,500 successful student participants from 97 countries and over 7,000 mentors from over 100 countries worldwide to produce over 50 million lines of code”. Through *Google Summer of Code*, accepted student applicants are paired with a mentor or mentors from the participating projects, thus gaining exposure to real-world software development scenarios and the opportunity for employment in areas related to their academic pursuits. In turn, the participating projects are able to more easily identify and bring in new developers. Best of all, more source code is created and released for the use and benefit of all.

(Source: [https://www.google-melange.com/gsoc/document/show/gsoc\\_program/google/gsoc2014/about\\_page](https://www.google-melange.com/gsoc/document/show/gsoc_program/google/gsoc2014/about_page))

enterprise intended to invest in the project and pool resources and capabilities together in order to meet their objectives. By doing so, the Shinken project became stronger and more technologically and economically viable and sustainable as they received funding, additional human resources, and target market feedback. The client saved a lot of money by doing shared R&D in public domain instead of incurring all costs of R&D in-house – while enjoying higher quality R&D outcome due to using the power of a community. The OSS firm managed to fill the technological gap that existed between client IT department and Shinken project while providing support guarantee for an extended period of time.

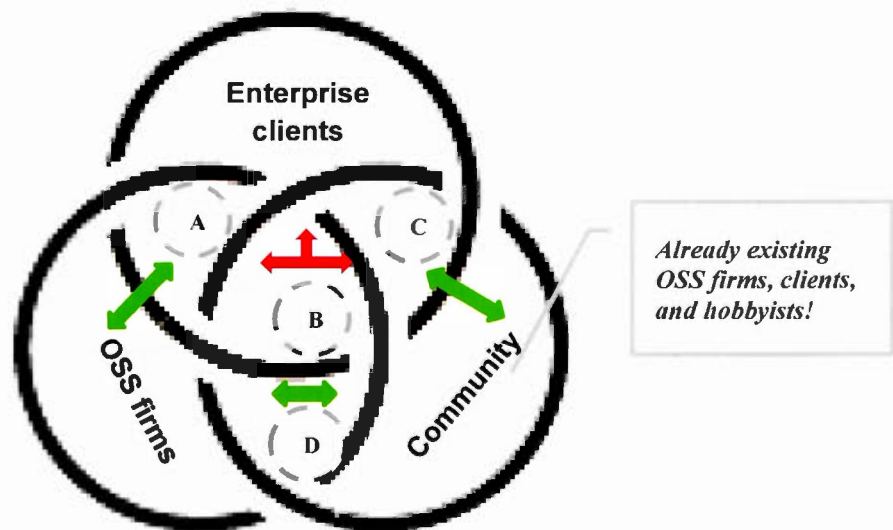
There are many more similar examples and cases in my formal and informal interviews, short discussions and observations that suggest coexistence and intertwining of interdependence and independence in OSS projects. However, what seems to be more important is the impact of this interwoven notion, and how we can interpret it in conjunction with other main and sub-categories discussed earlier. Finally, the concept of interdependence-independence can be schematically represented by the ‘augmented view’ of Borromean Links (Figure 5.16), as it is emerging from the interviews.

Figure 5.16 is built upon the conceptual subcategory of “relationship” and the relationship typology it discusses: i.e., dyadic vs. triadic relationships. It represents the ‘augmented view’ of Borromean Links because it also discusses the nature, type and role of relationships. On a more specific note, all three rings (enterprise client, OSS firm, and community) are interdependent despite being independent actors. If one disconnects, the other will do so (i.e., representing core concept of unity).

However, the zones are related through different kinds of relationships. The three zones (A, C, and D) host the dyadic relationships. These relationships highlight

independence of one actor group from the other. They also emphasize the duality of actors' technological collaborations in OSS RDIP. "Zone B", on the other hand, hosts triadic relationships among three actor groups in which case the three actors' resources dependency, integration and direct collaboration seem to be in motion. Overall, irrespective of actors' engaging in either dyadic or triadic relationships, the notion of interdependence represents the *conceptual glue* and reflection of a reality that all three actor groups are an indispensable part of the collaborative OSS RDIP. Each brings in a specific set of capabilities and skills, plays a particular role, and complements the other two in this process.

Figure 5.16. An Augmented View of the Borromean Links Model of OSS Development



*Source:* Author has adopted the basic concept of Borromean Links model from the literature, and augmented the conceptual relationships based on the primary data gathered from interviews and observations.

#### 5.3.4 Sustainability

Interviews with a wide variety of the OSS actors and secondary data with regards to participant's role in the open and collaborative innovation process, their vision and perspective, attitudes and motivations, resources allocation, all point towards the sustainability of communities and OSS technology development process as being essential. In fact, what distinguishes pre-dominantly OSS from closed software is its *ability to live and thrive in the public domain*—experiencing a dynamic evolutionary life which makes the software technology, its development process and its growth a sustainable phenomenon. Obviously, 'openness' and 'flexibility' associated with OSS development process allow resources and knowledge to flow among actors in order to enrich the RDIP.

As a case in point, OSS firms economically enjoy sustainability through providing their supports and expertise long-term. One OSS firm's department head mentions about the underlying reason to keep investing in community of OSS projects as: "*To be able to offer [our clients] support in the long term*". Client enterprises also benefit from sustainability and view it at the spearhead of their efforts due to different reasons earlier touched upon. For example, one large client corporation's IT department head emphasizes that getting a newly written piece of code (e.g., a new feature which performs a particular function for the client's operations) integrated into the core code in public domain has long-term benefits. Once the code is finally accepted and integrated, it will be maintained by the community of developers and evolve in harmony with other dependencies. The net result of this harmonious evolution in the commons is that the client's IT department frees some of its limited human resources from being worried about the sustainability of the feature they are relying on.

*"When it eventually gets integrated into the [OS] platform, then I don't need to maintain it anymore...like for the next 10 years so the modules that I have added to this OSS gets integrated into the normal releases then I don't need to*

*maintain it anymore; because it's either maintained by the community or the provider that's behind it. That's a benefit for me because it benefits the community but for me at some point these things gonna evolve on its own and I won't need to maintain it anymore. Since we have limited resources the more others can do the better it is for us because we can work on other stuff."*

(IT department head of publicly-held enterprise client)

Project leaders' (administrators/maintainers) efforts to manage diversity and lead people as a cohesive unit towards achieving the projects' milestones also suggest projects' and collaborations' sustainability as being essential. For example, the concept of *dual-purpose leadership* bears upon issues like effectively managing technology, people, their requests, and resolving the conflicts in order to glue efforts together and reach the project objectives. This versatility in leading projects is critical to avoid unnecessary project forking and nurturing the group synergy. These examples are just a few to emphasize the contributing roles of other categories and subcategories towards promotion of the key concept of sustainability. In order to further reduce abstraction associated with the core concept of sustainability, I propose to break it down into four levels: 1) *Technology* level; 2) *Resources* level; 3) *Relationship* level; and 4) *leadership* level.

The *technological* level encompasses two concepts of interoperability and maintainability associated with OSS. It hinges upon the inherent characteristic of software which is "*modularity*".

Modularity of software coupled with openness of the OSS development methodology allow and enable many developers to get involved in collaborative software design and production. However, as OSS evolves rapidly<sup>75</sup>, it is critical for users to remain

---

<sup>75</sup> A quote from an enterprise clients' IT department shows rapid software evolutionary path. "Sometimes the OSS project changes often. Its development is very fast, and sometimes it is very difficult to reuse the same application because the API has so much changed that it is more difficult to integrate it. So sometimes the OSS projects are moving so fast that it is very difficult to integrate the

(constantly) connected with the project they are using or building upon in their in-house R&D projects. Remaining in touch is captured through concept of “relationships” and ensures that the underlying software remains interoperable among all platforms. Further, software that is not maintained becomes buggy, and ineffective. In long term, if an OSS that is not well maintained, it will become technologically irrelevant to its commercial and technological environment. It does not inter-operate harmoniously with other dependencies and libraries and it loses its capability to meet market needs. Therefore, maintainability and interoperability go hand in hand and ensure the sustainability of the OSS from a technological perspective.

In addition, to maintain a project so that the software remains interoperable and functioning, actors need to allocate tangible and intangible resources on a sustainable basis. Thus, the *resources* level concerns the continuous tangible and intangible resource allocation to maintain the projects. Bilateral and trilateral knowledge flows, engineering expertise, end-user feedback, software testing results and quality assurances feedback are all essential resources for project survival. Furthermore, sharing the cost of R&D through crowd funding, and donations provide the financial fuel to satisfy extrinsic motivations of developers to keep coding and remain active for longer time periods on the project. The following quotes explains role of QA and resources sustainability.

*“A lot of challenge with open source software is the quality of software itself. I call this end user experience. It is the QA (quality assurance) process. How can you test it? When the commercial company takes your open source project and embedded into their own product and ship to their customers, it will go through their QA teams. They will find bugs that you would never be able to find. If you make available professional services and that commercial company can engage you as one of the developers of the project, for helping*

---

new version. So, it is difficult to use an OSS project because we know that it won't be the same in one year. Or the project won't live after one or two years. So it is important to know if the project lives and won't disappear.”



*them to commercialize or embed your project then not only you will earn revenue from that, and you must be able to earn some revenue from an open source project. An open source project with no revenue is not going to be viable in the long-run. A lot of times when you create an open source project from scratch, it's gonna be hard to sustain unless there is an ecosystem in which commercial companies can build upon and contribute back to the software... So what happens next is even if they ship and nothing else happens, the product has gotten more stable along those bugs fixed in the core, developers around the product has earned revenue, and you have a design win. You have reached a milestone in your open source project where a commercial company has embedded your software into their product."*

(BigBlueButton project administrator)

*"We share the effort because there is no way one can make all the effort alone and provide all what the customers ask; and there is no way we can charge everything to one client. Yes. This might be purely R&D and I need to finance it. .... There is no way that one single company invest alone in this project and finance the full amount... because no company is sure on the return on its investment."*

(Odoo project leader)

*"I just don't believe that unless you are getting economic benefit from working on an open source project that the open source project succeeds in the long term. There has to be an ecosystem around it where there is exchange of services and revenues generated. And of course this revenue goes back to development through funding developers or compensating developers' efforts. Linux is doing so well because it has such a strong vibrant ecosystem. most of the committers are commercial companies who are benefiting economically from improvements to Linux."*

(OSS project leader)

The third level is the *relationship* level. It includes the existence of dyadic as well as triadic relationships. Together, they ensure resources mobility and therefore interoperability and maintainability as the end result. Therefore, sustainability of OSS projects, the community around them and the software itself hinges upon inter-relationships as they are the vessels and veins of the OSS ecosystem through which



tangible and intangible resources flow fluidly. The following quotes illustrate the critical role of relationships in the innovation process and maintain quality of software development. Actors look at the relationship building as a strategic move as they know without them developing and maintaining high quality software is nothing but just a fantasy.

*"I think the quality of our products could be worse if we cut our relationship with open source. It would definitely need community... We learn from it and we try to build something better with it."*

(OSS developer)

*"We have also this strong relationship with the community. We are like a gateway for anyone, for any company to reach their relevant community member. For example, last week the customer wants a feature that unfortunately I don't have the resources internally to provide it; and it would take a lot of time to explain the project and train one developer to be able to provide the feature. So I just contacted a few friends from the community on the web, and they told me they could provide it in one day! So two days after, I got the module providing the feature. So I skipped the learning curve, I skipped the availability of my team and my resources but I am still able to provide the service through my relationship with the community..."*

(Odoo project leader)

*"I work on strategic long-term relationships; and I farm those relationships. I have known these people for many years and I go sell strategic stories on why and how we can do it."*

(OSS firm department head)

*"So we look at it like a strategic long term relationship rather than trying to oversell at the start. My predecessor had that problem. He wanted to oversell at the start and this scared the clients away....So this is a perfect model, we train you and teach you the expertise to help yourself. We go to the embedded part to get them up and going."*

(OSS firm sales manager)

Finally, concept of dual-purpose leadership helps putting all pieces of the puzzle together in harmony. It enables us to realize the actors' roles in a complementary fashion and depict the big picture they are meant to paint. Leadership glues these pieces together and ensures they remain connected, as time passes by. Thus, the leadership level emphasizes alignment of all other three levels together effectively to ensure sustainability.

#### 5.3.5 Success

The concept of SUCCESS is a very fuzzy notion. Even when we consider interdependence-independence of actors, governance structures, leadership, and sustainability, there can be differences of opinion in OSS context about success. It means different things to different people, and different actors measure it differently.

It is a multi-faceted notion with several factors contributing to it. In this research, success has emerged as a concept which has two main components: *technological* side and *commercial* side. Most of the subjects in this study view the two as complementary to one another. Thus, the two, together, build up the success category.

*Technological success* is perceived as the quality of software code, its maintainability and interoperability, as well as the quality of technology management process. In fact, a successful OSS tool is viewed as one that functions well, performs solidly, and fulfills the technological expectations (i.e., a functionalist perspective).

The quality of software code is very important from the functionalist perspective. The software that is buggy, and that it crashes frequently cannot be considered reliable and trustworthy. Issues like long-term interoperability and maintainability are also conceptual ingredients of technological success. At one point, even if there is an OSS tool which is technologically superior to a competing counterpart because of quality

of coding and packaging, it may not become a technological success if it lacks the interoperability feature and if it cannot be matched with other software tools– an indirect quote from an OSS software developer. Technologically successful OSS tool is, thus, a relevant tool that fits the larger body of tools. In brief, a piece of code, tool, or an application that functions in connection with others.

There is also the management issue associated with technological success. For example, there may be an OSS tool which is highly interoperable with other tools and may even be popular but since the community of developers around the software is not technologically vibrant or not managed properly, then enterprise client users tend to shy away from incorporating that tool into their information systems' infrastructure. The issue is that the technology cannot be trusted because the community around it is a questionable one. One interviewee about SugarCRM mentions that:

*“From the ones I have seen fall off the chart is because they weren't managed properly... Like Sugar CRM. It was a very popular technology and it was against sales force. But then the code became like a spaghetti soup you know like too many chefs and not enough management on top of that. So the developers couldn't develop what they wanted, and it got messy and the developers' time got wasted and it became frustrating and the community directions were bad...In fact, managing the community matters and the community has to police itself. You see all those successful communities they are managed, maintained and they have described selected road map. They do all these things that require a structure. Community takes a lot longer because it is a community approach but they continue on. They sometimes have to adapt...”*

(OSS firm sales manager/OSS evangelist)

On the other hand, *commercial success* is a function of four main factors: a) *high commercial viability* (market demand); b) *diversity in user base* (enterprise clients,

OSS firms); c) *financial status* (revenue model and profit generation); and d) *project leadership*.

High commercial viability is reflected in market demand for a particular OSS technology. When a project is commercially successful, its OSS technology gets integrated into professional enterprises' software tools and products. This is what some interviewees have called a "design win".

*"I would argue that all of the successful Open Source Projects have very strong commercial interests where the commercial company see the economic benefit of having a strong open source project."*

(BigBlueButton community leader)

For example, Odoo (formerly known as OpenERP) is an OS enterprise resource planning software (ERP) which has been in market demand for several consecutive years since its inception enjoying an upward trend of adoption by a variety of enterprise clients. Despite having technology competitors from both proprietary (SAP ERP, Oracle E-Business Suite, etc.) as well as open source (ADempiere, OFBiz, Openbravo ERP) worlds, Odoo has successfully served many client enterprises worldwide. Projects with commercial viability are those capable of responding to real market needs.

*"OpenERP has raised \$10 million dollars in funding to support its R&D efforts and commercial growth. As part of the growth strategy OpenERP is rebranding to Odoo, to better reflect its expanded areas of focus in CMS & Ecommerce and POS applications beyond the core ERP function. Odoo has improved its pricing and included a free version for up to two users for very small businesses to enjoy a completely integrated suite of business solutions."*

(PRNewswire, May 16, 2014, San Fransisco)<sup>76</sup>

---

<sup>76</sup> Source: <http://www.prnewswire.com/news-releases/openerp-becomes-odoo-and-closes-10-million-in-funding-259488771.html>

Furthermore, commercially successful OSS projects are those which gain diversity in their user base, meaning that enterprise clients from different industries are keen to use the tools for different purposes. The diversity of user base means injecting the community of developers with different types of valuable end-user feedback with regards to functionality matters and testing the software in real life situations. These valuable knowledge flows have also positive impacts on technological success of the project as they contribute to quality enhancement measures and further interoperability of the software itself.

Moreover, majority of subjects view commercial success as a matter of financial status of the project— therefore adding an economic perspective to it. More commercial viability means more companies are willing to invest in collaborative R&D and project maintenance. Every project needs to have certain number of highly professional developers, or as some call them “*star code writers*”, in order to take care of advanced coding requirements and architecture design. So their work is not purely altruistic and they need to be *reimbursed* for part of their efforts through OSS project funds. Therefore, a project that is commercially successful is one that has a revenue model in place to be able to keep those star code writers in the project and improve professional aspect of the technology and serve clients professionally.

*“An open source project with no revenue is not going to be viable in the long-run. A lot of times when you create an open source project from scratch, it's gonna be hard to sustain unless there is an ecosystem in which commercial companies can build upon and contribute back to the software...”*

(BigBlueButton community leader)

*“Unless the community is based on a 'revenue-driven model' which acts as the community and as the enterprise like SFL. Liferay is a good example. They are community, there is an open source software involved but they charge service fees. They are not in a non-for profit situation. So you can't say they are purely community when they are charging money....”*

(OSS enterprise client, financial sector)

Finally, the rhetorical question that some interviews have highlighted along their discussions concerns: *Who is going to take care of commercial aspects of an OSS project?* Based on the interview data with commercially viable and thriving OSS project leaders, I have come to realize that effective leadership is not the one that only focuses on technological issues, but also it is the leadership that has “business sense” and knows about private sector organizational issues like branding, marketing, quality assurance to ensure client satisfaction, etc.

*“The goal is stability, usability, and features, in that order. So if the feature is not stable, it doesn’t matter how useable the features are....It is not just possible to do a BBB if you had many people making many contributions. The challenge for our software is not so much the development, it’s the testing...and very tight control over the quality of the product....Well, once we, the committers, merge something into the core or to master, take on the responsibility to support it...So, contributors will come and go, but they are really contributing to the things that are on our roadmap. Because once it goes in, we can’t rely on the external developers to always be around...we as the committers basically have to make sure that whatever goes in is solid...”*

(BigBlueButton project leader)

*“So you can write a lot of software and if doesn’t...how about good brand... If I look at it from marketing point of view; a brand is a promise of some benefit and the delivery of that benefit. So in BBB we promise stability and ease of use. And if we don’t deliver that then there is a negative brand...we release on quality, not date...”*

(BigBlueButton project leader)

All in all, technological and commercial successes together build the concept of OSS project success where the two key components are interlinked and complementary so much so that without one the other cannot be realized.

#### 5.4 Putting categories in perspective: The core propositions and the theoretical model

Having analyzed each conceptual category in detail, I have managed to shed light on the black box of their related and interrelated conceptual building blocks including their overlap with other categories, and subcategories. This effort has enabled me to classify my findings of this chapter under two interconnected subsections: a) *The Core Propositions*; and b) *A Theoretical Model*. The propositions developed here encompass the interrelationships among concepts and explain how they make sense in relation to one another. These propositions further touch upon several existing conceptual overlaps. The theoretical model, on the other hand, presents *schematically* how the independent, moderating, intervening, and outcome variables are related and influenced by one another.

##### 5.4.1 Developing the core propositions

Table 5.7 presents the four core propositions which form the inductive theory of OSS technological collaborations developed ground up in this research. It aims to clarify each main and subcategory, and explicate how the categories are related to one another. More specifically, propositions focus on the connection between relationships, leadership role, and contextual factors and their impact on the outcome constructss: sustainability and success of technological collaborations and RDIP and explain them in more detail. In follow-up, Figure 5.17 schematically demonstrates how the categories are linked.

Table 5.7. An Inductive Theory of OSS Technological Collaborations:  
The Core Propositions

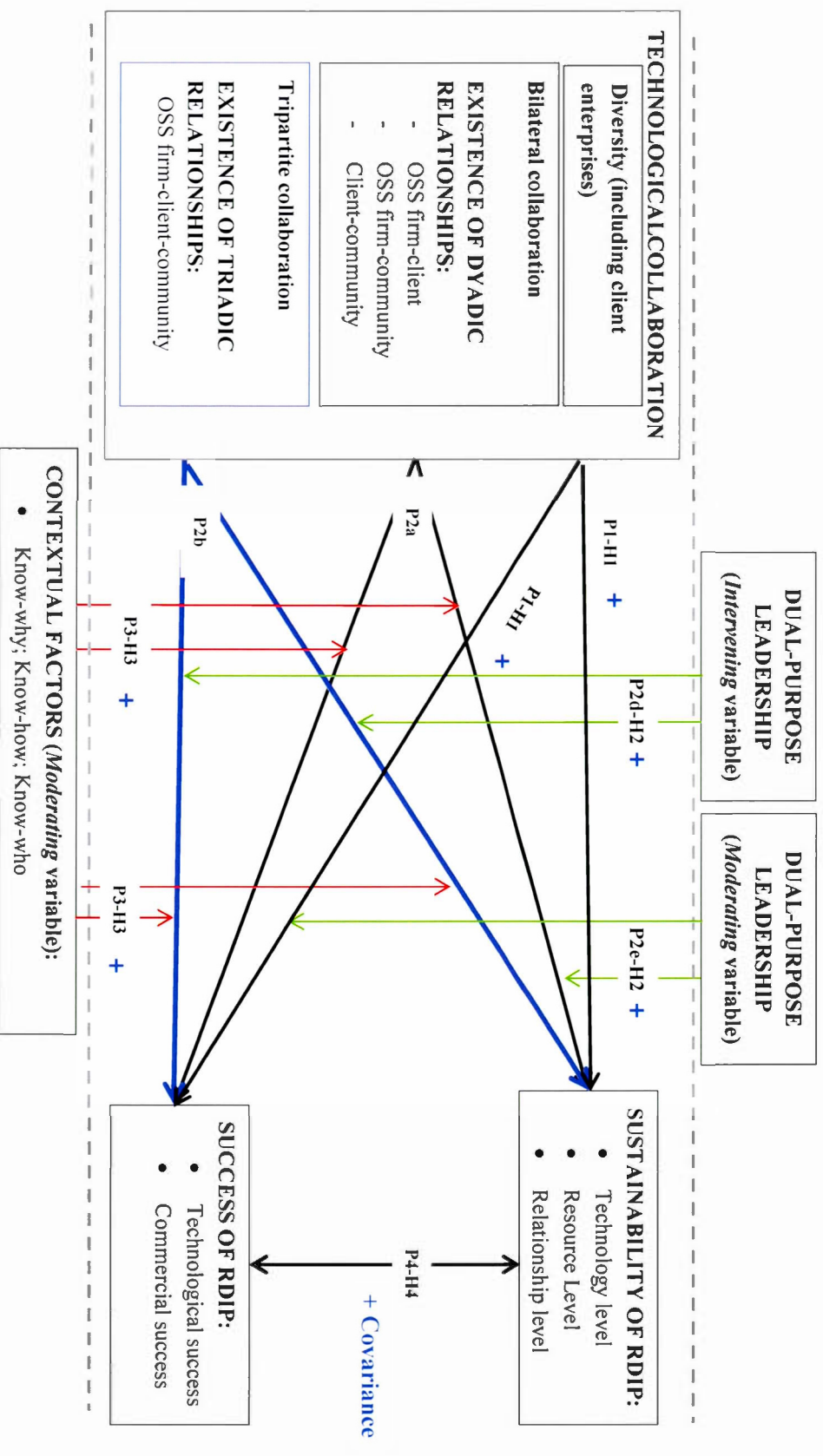
A theory of successful and sustainable OSS tech. collaborations: Core propositions
<p>Context: Composition of actors and their diversity within OSSTC</p> <p><b>P1.</b> The diversity of actors (including enterprise clients) – under certain conditions (i.e., context for collaborations: know-who, know-how, know-why)– is likely to positively influence both the sustainability and success of collaborative OSS RDIP.</p>
<p>Context: Technological collaborations and typology of relationships</p> <p><b>P2a,b.</b> Existence of either <i>dyadic</i> or <i>triadic</i> relationships is likely to enhance the success and sustainability of OSS collaborative RDIP, and projects that host them accordingly.</p> <p><b>P2b.</b> Having observed that triadic relationships include more pecuniary and non-pecuniary resources, diversity of (tacit) knowledge and capabilities; they tend to be more effective in enhancing chances of project success and sustainability.</p> <p><b>P2c.</b> Having observed that ‘<i>tripartite collaborations</i>’ involve more diversity of incentives, there is more chances for conflicts to arise; and, it is more likely for disagreements to lead to dissolution of projects.</p> <p><b>P2d.</b> ‘Dual-purpose leadership’ tends to play an <u>intervening</u> role on the relationship between tripartite collaborations and success and sustainability of RDIP.</p> <p><b>P2e.</b> ‘Dual-purpose leadership’ tends to play a <u>moderating</u> role on the relationship between dyadic collaborations and success and sustainability of RDIP.</p>
<p>Context: Contextual factors’ impact on success and sustainability of RDIP</p> <p><b>P3.</b> Contextual factors (2 Ks) are likely to indirectly influence the success and sustainability of RDIP. Put simply, contextual factors tend to enhance the quality of technological collaborations and ensure further goal-oriented developments.</p>
<p>Context: The interplay between success and sustainability of OSSTC and projects</p> <p><b>P4.</b> In view of the particularities, vulnerabilities of OSS technology, and empirical data inputs (observations), there seems to exist a strong interdependence between success and sustainability constructs. Therefore, they tend to influence one another and covary.</p>
<p>Source: The author.</p>



#### 5.4.2 The theoretical model: Identifying the key factors influencing the success and sustainability of collaborative OSS RDIP

The theoretical model (Figure 5.17) is one of the main outcomes of this research project. This model is built ground up –i.e., inductively– based on primary data collected through observations and interviews. It comprises five core conceptual categories and it aims to demonstrate how the typology of relationships (dyadic vs. triadic relationships) among actors influences the sustainability and success of collaborative OSS R&D and innovation process. Thus, the two conceptual categories of “sustainability” and “success” form the *outcome variables* of this theory. The propositions presented in detail in the preceding subsection explicate the interrelationship among the conceptual building blocks of this model.

Figure 5.17. The Theoretical Model of OSS Technological Collaborations



## 5.5 Discussions

Past research (Weber, 2004; Hang et al., 2005; Kelty, 2008; etc.) has tried to understand OSS collaborations and collaborative software methodology. However, their approach and focus have been different from those of this study.

Kelty (2008) focuses on *free* software to approach collaborations and coordination. He focuses on Minix vs. Linux and describes Linux success due to accepting almost everything and having no goals (Kelty, 2008, pp. 218-219). He claims that “coordination in free software privileges *adaptability* over planning” (pp. 218-219). He sees adaptability as a process that “involves more than simply allowing any kind of modification; the structure of FS coordination actually gives precedence to a generalized openness to change, rather than to the following of shared plans, goals, or ideals dictated or controlled by a hierarchy of individuals” (Kelty, 2008, p. 211). He also stresses the concept of need-based contributions or need development and views it as an organic process. In short, Kelty’s arguments ignore the distinction between Free and OS software, downplays the role of strategic planning in collaborative software development, and while underlying need-based contributions, he views that as an organic rather than a managed process.

On the contrary to Kelty’s approach and focus, this study distinguishes between free and open source and mainly focuses on commercial open source software tools and projects that will be integrated into a variety of information systems in a wide range downstream industries— i.e., a commercial business case software. Hang et al. (2005) also studied OSS as a business opportunity and not merely an ideological phenomenon.

The themes and processes emerging from the data (e.g., dual-purpose leadership—Table 5.6; seven stages of development from the Eclipse Foundation—Figure 5.14;

etc.) suggest the significance of professional and rigorous project management and calculated technological collaborations in order to ensure success and sustainability of the projects and collaborations . For instance, two in-depth case studies of 'Tiki Wiki' (OS content management system) and 'BigBlueButton' (OS web conferencing system) show how managing technology development process, forecasting the future technology trajectories, and effectively leading the OS goal-oriented networks can create positive network externalities and sustainable OSS development process.

Furthermore, involving large number of people is not the only solution to responding to the needs. In fact, in order to understand needs we need to understand the demography of actors engaged in software development. Weber (2004, p. 71) also underlines the shortage of data on demography of the OS communities and emphasizes the relative importance of any particular contribution or even the level of effort that any individual puts in. Indeed, it is not always about the quantitative measures. Rather, qualitative side matters too. For this and many other reasons, the interviewed OSS firms and community leaders have emphasized the role of enterprise clients in the collaborative RDIP (see Figure 5.11. OSS Enterprise Client Typology). Just imagine a community primarily dominated by hobbyists. They do not necessarily have end-users to account for. This means their needs are more passion or personal utility driven. However, in case of a financial organization or a transport company, as a user of OSS, there are serious expectations of smooth functionality downstream. Taking this functionalist approach to software development and using the pool of talents from the commons requires a strategic client need management system (See Figure 5.9. Linear Built-To-Order Collaborative R&D and Innovation Process) .

Moreover, some researchers (e.g., Weber, 2004) does not deal with the issue of pecuniary compensation or economic/profit side of OS projects in detail. Nor he correctly sees the reasons for heavy reliance on the commons. Weber (2004, p. 75)

states that “a good programmer is lazy like a fox” who is “always searching for efficiencies”. His argument is, writing quality codes is a daunting task which takes time, and since these OS programmers do not get “directly paid” for their time spent on coding, they have “a particularly strong incentive” to use the OS commons— i.e., already existing libraries and tools. I partially agree with this argument and in fact some of these claims emerged from my findings as well (albeit not in those exact words).

However, two major arguments are missing here. First, as the analysis of cases of BBB and Tiki Wiki communities reveal, commercial or economic sustainability is central to sustainability of the OSS RDIP. Relying on commons is not only due to developers’ laziness or the logic of avoiding the reinvention of the wheel. Developers are motivated to save time and avoid reinventing the wheel in order to reduce the costs of OSS R&D and increase the margins. Many of professional OSS developers are actually coming from public and private enterprise. For them, using the commons is a way to save cost.

However, saving costs of software R&D projects explains for part of their reliance on the OS commons. Other reasons, such as ‘creating positive network externalities’, ‘influencing technological trajectories’ in ‘the direction’ they find appropriate for their future downstream products, and ensuring ‘maintenance’ and ‘interoperability’ of their information systems long term are among the other reasons to act like “a lazy fox”.

Based on these observations and primary data, I therefore can complement and progress the core propositions (Table 5.7) and the theoretical model (Figure 5.17) by proposing the following testable hypotheses.

*H1. Inclusion of enterprise clients— provided that they are actively engaged—early on in development process has a significant and positive impact on technological and commercial success of OSS projects.*

*H2. The existence of a dual-purpose leadership tends to positively and significantly influence the success and sustainability of RDIP.*

*H3. Contextual factors are likely to have a moderate effect on the relationships between actors and their collaborations and success and sustainability of RDIP.*

*H4. There seems to be a covarying effect between success and sustainability of OSS RDIP, so much so that if there is an increase in one, the other is positively influenced and will tend to be increased as well; and vice-versa.*

Hypotheses three is particularly important because we cannot only discuss why developers should engage in collaborative development but also consider how they can do so. Collaboration is empowered by, and, takes place within a ‘specific context’. Such context is identified by three main elements: ‘know-why’, ‘know-how’, and ‘know-who’. Know-why refers to the logic underlying the collaboration. Know-how captures the necessary technological capability (both level and intensity), relationship building and maintaining capabilities. Know-who refers to knowing the key actors with whom collaboration is to be initiated and maintained in order to engage in goal-oriented interactions. Each element has both ‘internal’ and ‘external’ consequences. Internal consequences refer to changes to be made internally in order to support the collaboration. For example, a client enterprise’s top management level needs to authorize IT department with further resource allocations towards tripartite collaboration. External consequences involve ramifications of each element on success and sustainability of OSS technological collaborations and projects. For instance, if adoption and integration of a particular OSS module falls under a long-term client enterprise IT strategy, then their continued collaboration ensures further sustainability of OSS project under focus. Overall, the three ‘contextual elements’, if

in place, seem to indirectly influence the success and sustainability of OSS technological collaborations and projects. More specifically, they seem to influence the 'quality of collaborations' captured through dyadic and triadic relationships and tend to moderate the connection between 'relationships' and 'success and sustainability' issues.

Finally, technological and commercial successes of OSSTC seem to be interlinked with two subcategories of sustainability; namely, technological and resources levels. In fact, the concept of technological success captures issues such as software interoperability and maintainability which also build technological level of sustainability. Concept of commercial success touches upon diversity of user base and revenue model of the project which tightly corresponds with resources level of sustainability. Therefore, it seems reasonable to conjecture that the two outcome variables of success and sustainability co-vary. Thus, since the two concepts of success and sustainability share similar conceptual building blocks while having their own differences, it is likely that the two outcomes co-vary. This means that when sustainability of an OSS RDIP increases, it likely increases its success with it; and vice-versa, when the success of the OSS collaboration enhances, it likely enhances its sustainability.

## 5.6. Conclusion

In this Chapter, I have explicated the character of OSS RDIP and technological collaborations through highlighting their main conceptual building blocks and showing how they are related and how they can influence one another. These details provide us with a finer-grained understanding of the internal workings of OSS innovation process and how each actor functions in relation to the others. In addition, I have clarified the interdependencies and interrelationships that exist among the key

concepts through development of four core propositions, four hypotheses, and a theoretical model. The main goal of this theory building process has been twofold: first, to identify, characterize, and detail each pivotal element of the OSS R&D and innovation process; and second, to clearly demonstrate how the main elements are interrelated. The ultimate objective is to ensure that OSS technological collaborations effectively lead to success and sustainability of OSS projects. Therefore, to this end, the propositions and hypotheses developed in this Chapter provide an explanation on how the key elements make sense when they are considered together. Finally, by way of unpacking abstract constructs and discussing them at a more tangible and operational level, I suggest that OSSTCs and OSS RDIP depend on some important contextual factors; namely, technological trajectories, network externalities, heterogeneity of actors which defy clusters in conventional sense, the rents which accrue to different kinds of actors (i.e., no Ricardian rents), and the uncertainties interwoven with path of the OSS technology development process. In the next Chapter, I intend to take a deeper look at OSS technological collaborations by way of comparing them with conventional strategic/technological alliances. This approach helps me to put the inductive theoretical insights developed under OSS technological collaborations in perspective and be able to bridge the crossroads between OSS and alliances literatures.



## CHAPTER VI

### FROM ALLIANCE FORMATION TO PLATFORM DEVELOPMENT: AN ALTERNATIVE PERSPECTIVE OF OSS TECHNOLOGICAL COLLABORATION

#### 6.1 Introduction

This chapter's main goal is to further theorize about OSS technological collaborations (OSSTC) and develop an alternative perspective which explains the platform view of OSS RDIP. To realize this goal, I first argue that OSSTC is different from conventional strategic alliances (e.g., joint ventures) and informal cooperation. In fact, they represent a 'new breed of animal' in the jungle of interfirm cooperation. Despite mirroring several traits of strategic alliances (e.g., using formal interfirm contracts as conduit for knowledge transfer), OSSTC are different from them as they often take place through informal, short-term, and trust-based relationships. Yet, OSSTC do not fall under the laissez faire informal cooperation because they are strategic, commercially goal-oriented efforts which are significantly influenced by organizational issues such as rigorous project management, leadership, motivational issues and reward system, to name a few. In addition, the nature of OSS technology (i.e., modularity, demonstrating characteristics of public goods, interoperability needs, evolutionary character, and network externalities) makes the case of OSSTC a unique one to conduct comparative studies. Thus, OSSTC is not the identical twin of either strategic alliances, or informal cooperation; they are simply a horse of a different color.

Moreover, earlier literature on OI has extensively discussed the ever increasing phenomenon of openness in association with BM, strategic management, firm's innovation performance, and has even shed light on rudimentary yet fundamental questions in regards to degree of openness and its consequences. However, a genuine OSS technology platform can neither be classified in terms of strict proprietary forms of platforms (e.g., Uber) nor can it be fully understood through the lens of OI where its core emphasis is on how opening up the innovation process can benefit or hamper firm-level innovation performance. Therefore, I believe OSS platform represents an Open Collaborative Technology Platform (OCTP) which may benefit from the doctrine of OI, but it deviates from the conventional thinking when it comes to one cornerstone of its strategic usefulness: i.e., long-term continuity and evolution. This central objective coupled with different nature and role of dual-purpose leadership developed in Chapter 5 lead us to view OSS technology platforms as OCTP where continuity is perceived as much critical as openness in the collaborative RDIP. Therefore, to develop and characterize OCTP, I identify seven dimensions, extract the relevant concepts from alliances literature and organize them around these dimensions. Further, I use the emerging concepts from Chapter 5 to distinguish OSSTC from conventional alliances. Finally, the characterization of OSSTC based on these seven dimensions form the theory of OSS OCTP.

## 6.2 OSS technological collaboration vis-à-vis strategic alliances concept

'Technological collaboration' acts as the engine of OSS RDIP, without which success and sustainability of OSS projects are at risk. It also relies on four main subcategories: 1) Context; 2) Code sharing complexities; 3) Relationships; and 4) Dual-purpose leadership. Further, each of these subcategories is comprised of several conceptual building blocks. For example, relationships in our model includes 'dyadic'

and 'triadic' types where each has its own issues, challenges, and requires a distinct set of capabilities.

Collaboration is a concept somewhat opposite to ownership, and mergers and acquisitions. It is in fact perceived as an '*alternative*' to both markets and hierarchies which lie at the heart of transaction cost economics, TCE (Williamson, 1979, 1981, 1985 1991). In short, in order to benefit from a product or a service, firms are faced with three main kinds of decisions: a) to go about it alone and make it in-house; b) to obtain the product or service through purchasing it from the market; or c) to realize their demands through jointly producing it with another partner or other partners (e.g., Das & Teng, 2000). Collaboration is about using the third option: i.e., co-production through partnership. This concept has been extensively studied in the context of strategic or technological alliances. For example, Niosi (1995, p. xi) highlights the significance of '*strategic alliances*' during 1980s and consider them as one of the most important "organizational innovations" in modern business world. There are several definitions of strategic alliances in the literature. For example, they can be defined as:

"...agreements characterized by the commitment of two or more firms to reach a common goal entailing the pooling of their resources and activities" (Teece, 1992, p. 19).

"purposive strategic relationships between independent firms that share compatible goals, strive for mutual benefits, and acknowledge a high level of mutual dependence" (Mohr & Spekman, 1994; as cited in Kale et al., 2000, p. 218).

"any independently initiated interfirm link that involves exchange, sharing, or co-development" (Gulati, 1995, p. 86).

“long-term contractual agreements between two or more enterprises aiming at development of new or improved product or process technologies” (Niosi, 1995, p. 3).

“...cooperative relationships driven by a logic of strategic resource needs and social resource opportunities” (Eisenhardt & Schoonhoven, 1996, p.137).

“...voluntary arrangements between firms involving exchange, sharing, or codevelopment of products, technologies, or services” (Gulati, 1998, p. 293).

“...voluntary cooperative inter-firm agreements aimed at achieving competitive advantage for the partners” (Das & Teng, 2000, p. 35).

Strategic alliances in the literature are approached through different terminologies. Hagedoorn (1993, pp. 371-372) views them as “strategic technology partnering”, in which sense it captures an “interfirm cooperation for which a combined innovative activity or an exchange of technology is at least part of their agreement”. He further emphasizes the “strategic character” of alliances in that the agreement between parties encompasses “the expected long-term effects of the agreement on the product-market positioning of at least one of the partners”. Therefore, two dimensions: “long-term perspective” and “contracts” become an indispensable part of the strategic alliances. Niosi (1995, p. 3) views a technological alliance (synonymously: technical alliances, collaborations, partnerships) as a “particular case of interfirm collaboration” that falls somewhere between “informal knowledge sharing” and “mergers and acquisitions”. He defines technological alliances as “long-term contractual agreements between two or more enterprises aiming at development of new or improved product or process technologies” (Niosi, 1995, p. 3). Thus, Niosi (1995) deliberately focuses on formal cooperation to the exclusion of “short-term, informal technological cooperation” while developing the theory of “flexible or cooperative innovation”. Furthermore, Das and Teng (2000, p. 34) view strategic alliances essentially as the result of “*resource integration among firms*” and formulate a general resource-based view of them.

In the present research, the ‘collaboration’ category leads to sustainability and success of OSS RDIP, and it reflects several attributes and characteristics of technological alliances. In fact, it acts as a conduit for transfer of tangible and intangible resources<sup>77</sup> such as: Knowledge and technological expertise, competences, learning and experiences, funds and donations, to name a few. However, it also differs from strategic alliances in that collaboration in OSS industry can be short-term, without formal contracts, and also it can involve a technological partnering between firms and individuals, firms and network of people, between individuals and foundations, etc. OSS technological collaborations also have a *strong social dimension* and this makes them different from the concept of “technological alliances” as studied by Niosi (1995). In addition, the ‘*technology roadmap*’ within OSS projects is not clearly defined and fixed; i.e., it can change as the surrounding technological dependencies change. Therefore, in order to develop an inductive and integrative theory of OSS RDIP, it is appropriate to draw on empirical and theoretical literature on alliances to highlight the points of dissimilarities between theory of OSS development and that of alliances.

### 6.3 Overlaps between strategic alliances and OI: The shared conceptual caveats

One of the main criticisms on OI framework rests upon the fact that the principles underlying closed innovation model have almost never existed and that they are misleading (for details see: Trott & Hartmann, 2009). Even a cursory examination of the past literature on strategic alliances highlights this point. Several theorists have already considered alliances as a *middle ground* between the firm and the market, highlighting the ‘*blurring*’ boundary of the firm (Lindsey, 2008; Baker et al., 2002; Garvey, 1995; Williamson, 1979; Richardson, 1972; Macaulay, 1963; Gomes-

---

<sup>77</sup> Wernerfelt (1984, p. 172) defines resources as “those (tangible and intangible) assets which are tied semi-permanently to the firm”.

Casseres et al., 2006) or what has been perceived as firm '*permeability*' within OI framework (e.g., Dahlander & Gann, 2010; Elmquist et al., 2009). This situation is further exacerbated mainly because the OI concept is principally received as "a firm-centric paradigm that is primarily concerned with leveraging external knowledge to improve internal innovation and thus the firm's economic performance" (Piller & West, 2014, p. 29). This firm-centric, economic approach towards innovation modeling pushes OI even further towards the theory of alliances as opposed to promoting the concept of 'open technology platform development' and 'distributed innovation networks'.

In fact, major factors underlying the very existence of strategic alliances (or sometimes worded as technological alliances, strategic interfirm partnerships and collaborations, etc.) which have been mentioned in the literature already include the very few erosion factors (see Chapter 2, Sections 2.2. and 2.6, for further information) and reasoning behind opening up the business models as mentioned by Chesbrough (2007b; 2003a, b, c) and even go beyond them.

Growing ubiquity of strategic alliances or interfirm collaboration has been emphasized in many important works in the past decades including, but not limited to, Hagedoorn (1993); Niosi (1995), Mowery et al. (1996); Grant & Baden-Fuller (2004). The seminal work of Hagedoorn (1993): "Understanding the rationale of strategic technology partnering..." provides a comprehensive overview of motives underlying formation of (strategic) interfirm technology cooperation. Hagedoorn (1993, p. 373) classifies the motives into three major categories: 1) motives related to basic and applied research and some characteristics of technological development; 2) motives related to concrete innovation processes; and 3) motives related to market access and search for opportunities.

More specifically, according to Hagedoorn (1993, p. 373)<sup>78</sup>, the first class of motives concerns: a) “increased complexity and intersectional nature of new technologies, technological synergies, ..., access to scientific knowledge or to complementary technology”; b) “reduction, minimizing and sharing of uncertainty in R&D”; and c) “reduction and sharing of costs of R&D”.

The second class involves: a) “capturing of partner’s tacit knowledge of technology, technology transfer, technological leapfrogging”; and b) “shortening of product lifecycle, reducing the period between invention and market introduction [time to market or TTM]”.

The third class of motives finally include: a) “monitoring of environmental changes and opportunities”; b) “internationalization, globalization and entry to foreign markets”; and c) “new products and markets, market entry, expansion of product range”.

Others (Grant & Baden-Fuller, 2004; Mowery et al., 1996; Niosi, 1995; Hamel, 1991; Kogut, 1988) have also shed light on other motives to form strategic partnerships. For example, in an attempt to develop a theory of joint ventures (JV)— as a form of strategic alliances— Kogut (1988, p. 319) mentions, firms may opt for JV as opposed to other modes such as licensing, or acquisition because it can act as an “instrument of organizational learning”. Therefore, firms form a JV in order to transfer the knowledge that is embedded in the organizational context which “cannot be easily blueprinted or packaged through licensing or market transactions” (Kogut, 1988, p. 319).

---

<sup>78</sup> Hagedoorn (1993) draws on a broad range of literature to build the three main categories of motives and form their subclasses. For the complete list of underpinning works please refer to the original work cited in the references section of this dissertation.

Hamel (1991) studies international strategic alliances and maintains that collaboration provides a means for a partner to internalize skills of the other and consequently to improve its position within and without the alliance. This highlights the issue of “inter-partner learning”. He further shows that not all partners are equally skillful learners and this asymmetry with regards to learning capability can change the balance of bargaining power between partners.

Niosi (1995, pp. 9-11) acknowledges the previous research on the factors which explain technological alliances and cooperation among independent firms and emphasize the most important ones as: a) “realizing R&D economies of scale”; b) “accelerating innovation”; c) “capturing the knowledge of users”; d) “reducing risk and uncertainty”; e) “coping with short life cycles of products”; f) “capturing other complementary assets”; g) “searching for standards”; h) “using new methods of management”; i) “responding to government incentives”; and finally j) “capturing regional externalities”.

Mowery et al. (1996), also mention about main motives underpinning alliances in terms of sharing cost and risks of innovation (Mowery, 1988); perceived shrinkage in product life cycles; coordinating and formulating technical standards and ‘dominant designs (Grindley, 1995); gaining market power (Porter & Fuller, 1986); acquisition of partner’s technological capabilities (Mariti & Smiley, 1983; Khanna, 1996). Further, Mowery et al. (1996) emphasize that firm-specific technological capabilities are often tacit knowledge and are subject to considerable uncertainty with regards to their characteristics and performance. Thus, they show that in some alliances, ‘absorptive capacity’ plays an important role in explain the extent to which technological capability has been transferred.



Grant and Baden-Fuller (2004, p. 62) further claim that viewing alliance formation as a “vehicle for organizational learning” has impeded the development of effective knowledge-based view of alliances. Therefore, to them, alliances are mostly advantageous because they allow partners to ‘access’ knowledge rather than ‘acquire’ it. Others have also made distinction between different modes of knowledge sharing. For example, Hamel (1991) emphasized the differences between gaining access to partner’s skills and internalizing them.

Thus, reviewing the literature on alliances with regards to major factors which underpin their formation and explain their existence leads us to *a major conclusion* that existence of strategic alliances properly explain the essence of arguments made by Chesbrough about “*why companies should have open business models*”<sup>79</sup> where he highlights: innovation inefficiencies, rising costs of R&D and shortening product life cycles as well as the possibility to enjoy open experiments (Chesbrough, 2007b). In fact, the major factors underlying the existence of alliances aptly capture Chesbrough’s justification for open business models.

Furthermore, motives to choose strategic alliances in lieu of direct ownership through market obtainment or mergers and acquisitions, as discussed above, is an implicit definition of the original conceptualization of OI: “*a paradigm that assumes that firms can and should use external ideas as well as internal ideas, and internal and external paths to market, as firms look to advance their technology*” (Chesbrough, 2003a, p. XXIV). This is exactly what alliances try to do. In fact, “openness” defined as: “*the pooling of knowledge for innovative purposes where the contributors have access to the inputs of others and cannot exert exclusive rights over the resultant innovation*” (Chesbrough & Appleyard, 2007, p. 57); lies at the heart of strategic alliances, otherwise how could the partners move along their technological collaborations if

---

<sup>79</sup>Chesbrough, H. (2007b). Why companies should have open business models. *MIT Sloan Management Review*, 48(2), 22-28.

they had reverted to concealing practices in partnership<sup>80</sup>. Gomes-Casseres, Hagedoorn, and Jaffe's (2006) study on the role of interfirm alliances as a conduit for technological knowledge sharing is yet another seminal work on the role of openness between partners and how they accrue benefits from it. Gomes-Casseres et al. (2006) reveal that the magnitude of the knowledge flows between alliance partners is greater than flows between pairs of non-allied firms. Therefore, it is no wonder that openness is needed and it helps. Certainly, it is not a magic when different studies have shown that alliances can increase the value of the firm, particularly if they include the transfer or pooling of technologies (see Chan et al., 1997). In addition, the empirical findings in the literature of technological alliances and their underpinning theories shed light on the fact that the 'six fundamental principles of OI' (Chesbrough, 2003a; See Chapter 2, Table 2.1) have been highlighted long before they have been represented by Chesbrough.

The 'main caveats' shared between conventional strategic alliances and OIP are that they both emphasize "accessing" or "acquiring" external knowledge and other sorts of resources and that they are extremely firm-centric. Yet, the emerging concepts and theory from Chapter 5 shows us a somewhat different reality about OSSTC and OSS RDIP. These two are not merely to 'access' resources, but they also aim at developing an '**open and collaborative technology platform**' (OCTP) which must remain accessible to participants in order to sustain over time. Eclipse platform is one such technology platform which not only facilitates knowledge and resources access but also ensures sustainable new knowledge creation— a point not so deliberately discussed in the aforementioned literature of neither alliances nor OI. Thus, in the following section, I will critically analyze the core theories explaining alliances, identify their gaps, use their strengths and lay the foundation of a specific theory the explains OSSTC as a distinct form of strategic alliances.

---

<sup>80</sup> See Kale et al. (2000) for a discussion on how partners share know-how and capabilities through strategic alliances while protecting themselves from opportunistic behavior.

## 6.4 Theories of strategic alliances and their implications for a theory of OSSTC

In view of breadth and depth of explanations on why firms should form strategic collaborations, the prior studies on alliances have gone well beyond the contemporary reasoning of OI. Further, literature on alliances is well-rooted in long established theories such as transaction cost economics, TCE (Williamson, 1975, 1985); resource-based view (RBV) of the firm (Wernerfelt, 1984; Barney, 1986, 1991); knowledge-based view, KBV (Grant, 1996); resource dependency theory, RDT (Pfeffer & Salancik, 1978), absorptive capacity (Cohen & Levinthal, 1990); relational contracting (Macneil, 1974, 1980); strategic behavior or strategic management (Porter, 1980a, b); agency theory (Jensen & Meckling, 1976); bargaining power (Bacharach & Lawler, 1981); to name a few. Table 6.1 presents a brief overview of five key theories used in the literature of alliances with the intention of highlighting the conceptual relevance of these with OSS technological collaborations and the OSS technology platform.

### 6.4.1 Transaction cost economics

#### Theory

Economics, defined by Lionel Robbins (1932, p. 16), is: “the science which studies human behavior as a relationship between ends and scarce means which have alternative uses”; and, from James Buchanan (2001, p. 29)’s perspective: “mutuality of advantage from voluntary exchange is . . . the most fundamental of all understandings in economics”. This understanding of economics would lead scholars like Oliver Williamson (2005) to focus on perceiving economics as the “science of exchanges” (Buchanan, 2001, p. 28). Therefore, ‘economic exchanges’ is at core of Transaction Cost Theory of Economics (TCE) where it offers a *framework to*

*understand a firm's decisions in regards to whether to conduct an activity internally or externally* (Williamson, 1979, 1981, 1991). TCE is comprised of two extreme governance modes: 1) markets; and 2) firms; where the theory's focus is on the trade-off between the two (Ozman, 2009). TCE is originally devised to provide an explanation on why essentially a firm should exist (Barge-Gil, 2013) in which sense TCE holds that firms exist to *minimize the costs associated with transactions* (Coase, 1937).

Within the doctrine of TCE the "transaction" is the basic unit of analysis (Williamson, 2005, p. 43) where transaction costs are of three major kinds: 1) *search and information*; 2) *bargaining and decision costs*; and 3) *policing and enforcing costs* (see Coase, 1937; also see Varian, 2002).

Coined by Ronald Coase (1937), the term transaction cost is used to develop a theoretical framework for predicting when certain economic tasks would be performed by firms and when certain others would be performed on the market. The classic example of this would be 'vertical integration' which is a transaction in which a firm makes the make-or-buy decision (choice between market and hierarchy). For example, a firm may decide to internalize some production functions, through for example, mergers and acquisitions, in order to reduce the transaction costs more effectively. This can be a cost-effective preference under the condition that an exchange's transaction costs (like purchasing from market) are higher than internalization costs.

There are two main underlying TCE assumptions in regards to the economic behavior of agents. Firstly, agents are rationally bounded; and accordingly all complex contracts become unavoidably incomplete (Williamson, 1998). Secondly, they are

opportunistic which implies that firms are formed for “attenuating the *ex post* hazards of opportunism by *ex ante* choice of governance” (Williamson, 1998, p. 31).

TCE, in comparison to neoclassical economics which views firm as a production function— i.e., a technological construction (Williamson, 1998), draws selectively upon three fields: law, economics, and organization theory, in an attempt to better understand the complex economic organization (Williamson, 2005). TCE is therefore concerned with the allocation of economic activity across alternative modes of organization (markets, firms, bureaus, etc.); it employs discrete structural analysis, and it describes the ‘firm’ as a ‘governance structure’ (Williamson, 2005; Williamson, 1998).

Following the TCE’s reasoning – which emphasizes a firm’s governance structure— Williamson (2005, p. 47) identifies three factors which have “pervasive ramifications” for governance (i.e., elements influencing the firm’s decisions). These factors are: 1) ‘asset specificity’ (e.g., tangible and intangible assets); 2) the ‘disturbances’ to which transactions are subject (and to which potential maladaptations accrue)— i.e. ‘uncertainty’; and 3) the ‘frequency’ with which transactions recur (which bears both on the efficacy of reputation effects in the market and the incentive to incur the cost of specialized internal governance) — i.e., ‘scale economies’. Further focusing on perceiving firm as a governance structure that is central to TCE, Williamson (2005) identifies three attributes for describing governance structure: 1) incentive intensity; 2) administrative controls; and 3) contract law regime.

TCE and strategic alliances

Since TCE spotlights transaction cost efficiency as the motivation for cooperation, several researchers have found TCE useful in understanding the phenomenon of strategic alliances (e.g., Das & Teng, 2000). Some have applied TCE's theoretical approach to the context of alliances to better understand the dynamics associated with their formation (e.g., Hennart, 1988, 1991; Kogut, 1988; Pisano, 1989; Williamson, 1991; Gulati, 1995).

For instance, the issue of 'partner's opportunistic behavior' has been under spotlight and mainly examined through theoretical viewpoint of TCE (e.g., Hennart, 1988; Kogut, 1988; Pisano, 1989; Williamson, 1991). The proposition is that a firm's concern about its partners acting opportunistically is likely to lead to higher transaction costs. To ward off this potential hazard in collaborations, TCE research has suggested firms to adopt appropriate contractual agreements or governance structures (e.g., equity JVs) to address these concerns. For example, according to Hennart (1988), equity control can reduce the problem of opportunism due to the JV's aligning of the partners' incentives. Similarly, Kogut (1988, p. 321) raises this issue of "mutual hostage positions through joint commitment of financial or real assets" as this leads to "superior alignment of incentives", and stabilizing "the agreement on the division of profits or costs". Pisano (1989)'s analysis of the motives for using partial equity investments (i.e., JVs) in collaborative relationships demonstrates that the use of equity linkages (as in biotechnology collaborations) is related to the potential transactional problems associated with transaction-specific knowledge, uncertainty, and small numbers bargaining conditions. Thus, due to partners' ex ante commitments to an equity alliance their concern for their investments reduces the possibility of opportunistic behavior over the course of the alliance (Pisano, 1989).

However, TCE has been challenged by some other researcher on several grounds. Zajac and Olsen (1993, p. 132) highlight TCE's weaknesses as: its "single-party analysis [as opposed to multi-firm orientation] of cost minimization"; and that TCE over emphasizes structural aspects (ex-ante/ex-post dichotomy vs. process). Thereby, the former leads to the negligence of the inter-dependence between exchange partners; while the latter ignores "processual aspects"<sup>81</sup> (Zajac & Olsen, 1993, p. 135).

Ozman (2009, p.43) further classifies the criticisms on TCE into three: 1) notion of opportunism which can be obviated based on mutual trust between partners (forming a network); 2) the need for a more dynamic approach to understand networks in environments of rapid change (network governance issues go beyond the dichotomy of firm vs. market); and 3) the lack of social processes in TCE.

More specifically in terms of strategic alliances, Gulati (1995) challenges the perspective offered by TCE concerning alliances as it uses "a singular emphasis on transaction costs" due to partner opportunism and advocates the use of contractual agreements or equity to resolve the issue. Such approach therefore gravely ignores the role of interfirm trust and the evolution of interpartner relationships (Gulati, 1995). Kale et al. (2000) aptly inform us that firms which can build "relational capital" in conjunction with "an integrative approach to managing conflict" will be able to solve the dilemma of learning from their partners through alliances while simultaneously warding off opportunistic behavior.

---

<sup>81</sup> Zajac and Olsen (1993, pp.136-137) argue that "any fundamental transformation in interorganizational exchange relationships over time needs to be understood primarily in terms of developmental processes, rather than a simple comparison of *ex ante* and *ex post* structural properties".

Next, Niosi (1995), in an attempt to develop a theory of alliances, casts doubt on whether the theory of transaction costs can explain the management of R&D. According to Teece (1989, p. 3-4; as cited in Niosi, 1995, p. 8), “cooperation is usually necessary to promote competition, particularly when industries are fragmented. Very few firms can ‘go it alone’ any more. Cooperation in turn requires interfirm agreements and alliances, [and] the boundaries of the firm can no longer be assessed independently of the cooperative relationships which particular innovating firms may have forged”.

Thus, in view of deficiencies associated with TCE, and in consideration of the fact that a strategic alliance is a particular case of interfirm cooperation where it lies somewhere in the middle between informal knowledge sharing and mergers and acquisitions (Niosi, 1995, p. 3), perhaps other theories can be more useful in understanding technological alliances. Das and Teng (2000), for instance, mention that since firms basically form alliances in order to gain access to one another’s valuable resources, RBV of the firm can be received as a plausible approach to further understanding strategic alliances; a perspective shared with several other researchers (Eisenhardt & Schoonhoven, 1996; Gulati, 1998; Rothaermel, 2001; van de Ven & Walker, 1984) who have viewed alliances as a quest for obtaining resources.

#### 6.4.2 Resource-based view

##### Theory

On the contrary to TCE (Coase, 1937; Williamson, 1975) which emphasizes the ‘cost minimization strategy’ either through markets or hierarchies; the Resource-based View of the firm, RBV (Penrose, 1959; Wernerfelt, 1984; Barney, 1986, 1991)



emphasizes the relationship between what a firm possesses, in terms of its resources, and what the firm achieves, in the terms of its strategic positioning vis-à-vis its competitors.

Barney (1991, p.102) develops the idea that “a firm is said to have a competitive advantage when it is implementing a value creating strategy not simultaneously implemented by any current or potential competitors”. This strategy is unique and generates rents for the original firm as owner of the resources, because the competitors may not simply possess these strategic resources. Thus, this stream of thoughts in strategic management literature has established a close link between resources<sup>82</sup>: “assets, capabilities, organizational processes, firm attributes, information, knowledge, etc.” (Barney, 1991, p. 101), and competitive advantage (Rumelt, 1984; Reed & DeFillippi, 1990) – where “a firm’s competitive position is defined by a bundle of unique resources and relationships” (Rumelt, 1984, p. 557).

Barney (1991) further characterizes these resources through a set of assumptions where: 1) strategic resources are heterogeneously distributed across firms; and 2) differences are stable over time (i.e., resource immobility/stickiness).

According to RBV, a firm’s resources lead to sustained competitive advantage when they have four key attributes: 1) valuable; 2) rare; 3) imperfectly imitable; and 4) there cannot be strategically equivalent substitutes for these resources that are valuable but neither are rare or imperfectly imitable (Barney, 1991, pp.99, 101, 105-106). Thus, RBV views a firm as a collection of sticky and difficult-to imitate resources (Penrose, 1959; Barney, 1986) which can generate firm rents through protection and deployment of them (Mowery et al., 1996).

---

<sup>82</sup> Wernerfelt (1984, p.172) defines resources as “those (tangible and intangible) assets which are tied semi-permanently to the firm”.

## RBV and strategic alliances

Having studied the rationales and motives underpinning the formation of strategic alliances (e.g., Hagedoorn, 1993), it becomes obvious that many firms essentially use alliances to either acquire or access valuable and strategic resources which cannot be obtained otherwise (see Das & Teng, 2000; Grant & Baden-Fuller, 2004). RVB does not deny usefulness of TCE in that when efficient market exchange of resources is feasible, then “firms are more likely to continue alone” (Eisenhardt & Schoonhoven, 1996), and rely on market. However, there are certain resources (such as tacit knowledge or know-how, organizational capabilities and routines) that are not perfectly tradable through market transactions; these resources may have been interwoven with other resources or embedded in organizations (Chi, 1994). Thus, a firm is faced with two options: mergers and acquisitions or strategic alliances. Also, “because the absorption of rivals is often too costly, in the permanent drive for corporate diversification, mergers and acquisitions are less common, and technical cooperation has increased” (Niosi, 1995, p. xi). Consequently, the inefficiencies associated with mergers and acquisitions and the unobtainability of resources through market transactions have made the quest for valuable resources through strategic alliances a favorable option. This has also brought RVB and its derivatives (resource dependence perspective, and knowledge-based view) under spotlight to better understand alliances.

Van de Ven (1976), quite early, developed the idea that the process of building inter-organizational relationships can be studied as a flow of resources among organizations. For example, a JV is formed when “two or more firms pool a portion of their resources within a common legal organization” (Kogut, 1988, p. 319).

Kogut (1988) explains alliance formation as a conduit for organizational learning; offering a third yet complementary rationale next to TCE and strategic positioning. In

his view, a JV serves as an instrument by which “firms learn or seek to retain their capabilities” (Kogut, 1988, p. 323). Thus, ‘acquiring the other partner’s organizational know-how’ and/or ‘maintaining an organizational capability while benefitting from another firm’s current knowledge or cost advantage’ remain as key rationales for forming a strategic alliance. In this scenario, organizational know-how is perceived as a kind of valuable resource which cannot be obtained through market transaction; thereby remaining unexplained by TCE.

Eisenhardt and Schoonhoven (1996) further adding to resource obtainment rationale introduce “strategic” and “social” explanations to explain alliance formation. They provide evidence that firms in ‘strong social positions’ or ‘vulnerable strategic positions’ tend to form strategic alliances. Indeed, TCE has remained ineffective in explaining strategic and social factors as propellers of firms entering into strategic alliances.

Although informative, RBV has been challenged by a few authors. For example, Priem and Buttler (2001) criticize RBV for not paying enough attention to the appropriate context in which the resources are more or less valuable; or how the resources can be obtained and how do they generate sustainable rent? Others, like Miller and Shamsie (1996), emphasize RBV’s lack of explanation with regards to when, where and how resources can be useful. Lawrence (1997) has also shifted the spotlight over the black box of processes through which particular resources provide competitive advantage. Further to these general criticisms, Das and Teng (2000) believe that RBV and its application to the context of strategic alliances are still under-explored and discuss how resource alignment between partners can directly influence collective strengths and inter-firm conflicts which in turn contribute to alliance performance.

### 6.4.3 Resource dependence perspective

#### Theory

Resource Dependence Perspective, RDP (Pfeffer & Salancik, 1978) stresses the notion of 'interdependence' among organizations based on the premise that no single firm possesses all the resources necessary to achieve its objectives. A basic consequence of this assumption is that interfirm relationships constitute a strategic response by which firms can control their dependencies and reduce the uncertainty which may arise as the result of their dependence.

Pfeffer and Salancik (1978) view interdependence as an indispensable attribute of social systems/social interactions and state that: "interdependence exists whenever one actor does not entirely control all of the conditions necessary for the achievement of an action or for obtaining the outcome desired from the action" (Pfeffer & Salancik, 1978, p. 40). Within organizational setting where no single firm can 'go it alone'; "Interdependence is a consequence of the open-systems nature of organizations – the fact that organizations must transact with elements of the environment in order to obtain the resources necessary for survival" (Pfeffer & Salancik, 1978, p. 43). Thus, "to understand the behavior of an organization you must understand the context of that behavior – that is, the ecology of the organization" (p. 1). Pfeffer (1987, pp. 26-27; as cited in Hillman et al., 2009, pp. 1404-1405) provides the essence of the RDP in view of the concept and concerning the interorganizational relations as:

"1) the fundamental units for understanding intercorporate relations and society are organizations; 2) these organizations are not autonomous, but rather are constrained by a network of interdependencies with other organizations; 3) interdependence, when coupled with uncertainty about what the actions will be of those with which the organizations interdependent, leads

to a situation in which survival and continued success are uncertain; therefore 4) organizations take actions to manage external interdependencies, although such actions are inevitably never completely successful and produce new patterns of dependence and interdependence; and 5) these patterns of dependence produce interorganizational as well as intraorganizational power, where such power has some effect on organizational behavior”.

Finally, Gray and Wood (1991, p. 7) mention that “the focus is on minimizing interorganizational dependencies and preserving the organization’s autonomy while recognizing that interorganizational relationships are necessary to acquire resources”.

#### RDP and strategic alliances

Researchers (e.g., Pfeffer & Nowak, 1976; Stearns et al., 1987; Goes & Park, 1997; Elg, 2000; Yan & Gray, 1994, 2001; Das & Teng, 2000) have applied RDP to the context of strategic alliances and have informed us better about the formation and termination of alliances.

Early research (Pfeffer & Nowak, 1976) supports RDP’s application to JVs finding that JVs are commonly formed between interdependent firms. Empirical evidence also supports the use of interorganizational relationships to reduce domestic and international environmental complexity and gain resources (Elg, 2000; Goes & Park, 1997; Stearns et al., 1987). Yan and Gray (1994, 2001) examine the balance of power between international partners, and find that alliances occur when organizations are mutually dependent but the partner controlling more important resources retains strategic control. Hallen et al. (1991) provide evidence that changes in partners’ bargaining power prompt interfirm adaptation, not necessarily instability of the alliance. Inkpen and Beamish (1997) find that as one partner accumulates key resources from the other, the venture becomes less stable. In other words, when the bargaining power’s balance between alliance partners shifts there is possibility for an

unplanned termination to take place within a JV (Inkpen & Beamish, 1997). This research puts under spotlight the the relationship between learning through a JV alliance and diminishing cooperation. In fact, when a partner acquires knowledge and skills, then inevitably motivation for cooperation diminishes. However, Niosi (1995) has emphasized that if alliances are *capable of producing 'new knowledge'*, as opposed to merely being a device to access and use partner's know-how, their survival will not be easily jeopardized by diminishing learning factor. This is mainly because partners are constantly motivated to learn from one another— a condition enabled by the existence and continuity of the alliance.

Others (Levine & White, 1961; Aiken & Hage, 1968) mention that firms tend to enter partnerships if they perceive critical strategic interdependence with other firms in their environment. In fact, Gulati (1998, p. 299) emphasizes a condition where one organization has resources or capabilities beneficial to, but not possessed by, the other. Richardson (1972), in a theoretical economic account, also proposed that the necessity for complementary resources is a key driver of interorganizational cooperation.

RDP has been an informative view of in that it explains why firms rely on certain resources which lie beyond their firm boundaries and possessed by other organizations. It also shows that managers resort to interorganizational relationships and manage them in order to exert control over their dependence and try to minimize them. It is a very useful framework to understand alliances and their interrelationships.

Like other approaches, RDP has also been challenged on several fronts. For example, in a review written by Hillman et al. (2009) about three decades after its invention, the authors mention that RDP has not been yet well integrated with other theoretical

perspectives to consider the dynamic nature of these dependencies and power as well as the multiplexity of interdependency (Hillman et al., 2009, p. 1408). Furthermore, more research examining multiple forms of interdependencies may be able to reveal of the conceptual perimeter of RDP (Casciaro & Piskorski, 2005). Lastly, more research considering the boundary conditions of RDT is needed (Pfeffer & Salancik, 2003, p. xxiv).

#### 6.4.4 Knowledge-based theory of organizational capability

##### The Theory

Knowledge-based theory (KBT) of organizational capability (Grant, 1996; Grant & Baden-Fuller, 2004), considers “knowledge” (emphasis on tacit knowledge or know-how) as the most important strategic resource of the firm. KBT rests upon four theory streams: a) competition as a dynamic process; b) RBV; c) organizational capabilities and competences; and d) organizational knowledge and learning. The theory holds that, in view of strategic importance of knowledge and since it resides in specialized form among individual organizational members, then it is plausible to emphasize “the essence of organizational capability” as to integrate individuals’ specialized knowledge (Grant, 1996, p. 375). Contrary to TCE (Coase, 1937; Williamson, 1985) which views a firm’s key role as to minimize transaction costs; KBT (Grant, 1996) highlights the key role of a firm as creating, storing, and applying knowledge (Kogut & Zander, 1992; Conner & Prahalad, 1996; Grant, 1996).

KBT has two major assumptions about success: 1) “under dynamic competition, superior profitability is likely to be associated with resource and capability-based advantages than with positioning advantages resulting from market and segment

selection and competitive positions based upon some form of ‘generic strategy’”; and 2) “such resource and capability-based advantages are likely to derive from superior access to and integration of specialized knowledge” (Grant, 1996, p. 376).

Following the KBT, there are two knowledge integration mechanisms: direction and routine. The former includes the process of codifying tacit knowledge into explicit knowledge embodied in rules and instructions (e.g., operating manuals) (Grant, 1996). Therefore, it is the principal means through which knowledge can be communicated at low cost between “specialists and the large number of other persons who either are nonspecialists or who are specialists in other fields” (Demsetz, 1991, p. 172). The latter, routine, has its own unique function mainly because not all tacit knowledge can be codified (Polanyi, 1967); and that any attempt to do so may be accompanied with substantial knowledge loss. According to Grant (1996, p. 379) “an organizational routine provides a mechanism for coordination which is not dependent upon the need for communication of knowledge in explicit form”. He further argues that following KBT, “the essence of an organizational routine is that individuals develop sequential patterns of interaction which permit the integration of their specialized knowledge without the need for communicating that knowledge” (Grant, 1996, p. 379).

Furthermore, KBT stresses the link between ‘sustaining’, “continual renewal of”, the competitive advantage as well as innovation and the new capabilities development. Such emphasis demands a firm; first, to extend their existing capabilities to encompass additional knowledge types; and second, to reconfigure existing knowledge into new capability types. The latter, as Grant (1996) emphasizes, seems to be more complex due to substantial costs incurred in terms of reducing the efficiency of knowledge integration; yet, it also perceived as even more important in gaining an edge.



## KBV and strategic alliances

Various studies view strategic alliances as means for sharing knowledge (know-how, technology, and organizational capability) and “organizational learning” (Kogut, 1988); and others have received them as means for *producing new knowledge*, particularly in high-tech industries such as biotechnology and advanced materials (Niosi, 1995). Hamel (1991) views collaboration (i.e., strategic alliances) as a means to internalize the other partner’s complex skills that are based on tacit knowledge. Alliances can short-circuit process of skill acquisition. Further, capacity to learn (i.e., absorptive capacity) as in case of internalizing partner’s skill can create a bargaining power for either of the partners (Hamel, 1991).

Mowery et al. (1996, p. 89) find that “equity JVs appear to be more effective conduits for the transfer of complex capabilities than are contract-based alliances such as licensing agreements”. They also emphasize the importance of absorptive capacity in acquisition of capabilities through alliances. Further, lower levels of transfer occur in unilateral contracts than in bilateral non-equity arrangements.

Distinguishing between knowledge generation and application, Grant and Baden-Fuller (2004, p. 77) argue that alliances serve as a means to access knowledge, and that they can “overcome the limits of firms in encompassing highly differentiated knowledge integration processes, while offering efficiencies in knowledge utilization”.

Dyer and Nobeoka (2000) make an attempt to find out about the ‘black box’ of knowledge sharing within Toyota’s network and successfully provide evidence that Toyota has the ability to effectively create and manage network-level knowledge-sharing processes so much so that it gives the company an advantage leading to relatively higher profitability. These results emphasize the role of networks in

creating a strong identity and coordinating rules. Based on their findings, a network seems to be more superior (effective) to a firm when it comes to 'creating', 'transferring' and 'recombining' knowledge. This is mainly because there is higher level of knowledge diversity within networks. Overall, Dyer and Nobeoka (2000)'s study spotlights 'dynamic learning capability' and 'speed of learning vis-à-vis its competitors' as two key factors which have led Toyota and its suppliers to gain sustainable competitive advantage. Such dynamic learning capability, to be effective, needs to extend beyond firm boundaries in order to create a competitive edge.

Overall, KBT has informed us of two main issues. First, specialized knowledge is a strategic asset that can lead to creation of competitive advantage. Second, to benefit from knowledge, firms must have the capability to constantly create, transfer, and recombine (new) knowledge; thereby extending their search for knowledge beyond their very own boundaries.

Within past research on knowledge sharing through collaborations through networks others have also raised three main dilemmas. The first concerns how to motivate self-interested network of members to participate in the network and to openly share valuable knowledge with other network members (Gray & Wood, 1991). The second involves the 'collective action' or 'free rider' problem (Dyer & Nobeoka, 2000). The third regards how to maximize the efficiency (speed and ease) of knowledge transfers among a large group of individual members (Dyer & Nobeoka, 2000). As the key to keeping the momentum, sustainability of competitive edge, is to constantly reconfigure existing knowledge into new capabilities, the last issue is of an exceptional value. Dyer and Nobeoka (2000) view creation of "multilateral ties among members (and a variety of processes for transferring knowledge)" as a solution to efficiency issue.

KBT also originally suffers from some weaknesses. Although it stresses the role of ‘flexibility of integration’ to gain sustained competitive edge, while highlighting the paramount role of networks in generating new knowledge, it has paid little attention to make explicit arguments over “how” relationships among actors (firms, individuals) facilitate knowledge integration. Furthermore, it also does not elaborate on the role of necessary capabilities such as absorptive capacity in integration process. KBT further does not pay attention to other types of knowledge such as “know-who” or “know-why” as facilitators of knowledge integration process.

#### 6.4.5 Relational view

##### Theory

At the heart of the Relational View (RV) of the firm lies the thesis that “a pair or network of firms can develop relationships that result in sustained competitive advantage” (Dyer & Singh, 1998, p. 675). This view of gaining competitive advantage is different from merely forming strategic alliances in that relational view emphasizes on interorganizational *rent-generating process* (Dyer & Singh, 1998) and the mechanisms which ensure sustainability of rents. In essence, the kind of relationship Dyer and Singh (1998) tend to emphasize is an ‘idiosyncratic’ one –as opposed to arm’s length relationships which are based on so called watertight formal contracts.

Therefore, the collaborating firms<sup>83</sup> are able to generate *relational rents*<sup>84</sup> through: 1) *relation-specific assets*; 2) *knowledge-sharing routines*; 3) *complementary resource*

---

<sup>83</sup> The collaboration can have as few as two firms and/or as many as thousands like the 23, 000 member banks of VISA organization.

*endowments*, and 4) *effective governance* to reach the stage of differential firm performance (competitive edge) and create supernormal profit; and also they are able to sustain doing so over time through: 1) *inter-organizational asset connectedness*; 2) *partner scarcity*; 3) *resource indivisibility*<sup>85</sup>; and 4) *the institutional environment*. In what follows, I briefly explain the tenets of RV from Dyer and Singh (1998)'s perspective.

*Rent-specific assets* are specialized strategic assets (see Amit & Schoemaker, 1993, p. 39) in which alliance partners heavily invest (Dyer & Singh, 1998). These assets are extremely valuable because they have a high degree of specificity and that they cannot be easily found in the market. Related to this issue, Williamson (1985) classifies asset specificity into three: *site specificity*, *physical asset specificity*, and *human asset specificity*. For example, a long-term rent-generating relationship's fruit may be development of a customized machinery tool (physical asset) or transaction-specific know-how (human asset) which cannot be generated otherwise.

The notion of *knowledge sharing routines* between firms has in its heart the fact that firms learn as they collaborate (for example, March & Simon, 1958) and is defined as "a regular pattern of interfirm interactions that permits the transfer, recombination, or creation of specialized knowledge" (Grant, 1996; as cited in Dyer & Singh, p. 665). In certain industries like biotechnology, this even becomes more critical in that those firms which are unable to establish learning networks are doomed to stay at competitive disadvantage (see Powell et al., 1996). Others have also emphasized the critical role of collaborative research especially in biotechnology mainly because it allows organizations to escape difficulties associated with knowledge internalization while letting them 'access' knowledge and human capital (Niosi & Hade, 1995).

---

<sup>84</sup> Relational rent is defined as "a supernormal profit jointly generated in an exchange relation-ship that cannot be generated by either firm in isolation and can only be created through the joint idiosyncratic contributions of the specific alliance partners" (Dyer & Singh, 1998, p. 662).

<sup>85</sup> Resource indivisibility is perceived as coevolution of capabilities (Dyer & Singh, 1998).

However, creating collaborative knowledge sharing routines is not exactly a walk in the park. This is mainly because firm's ability to exploit external knowledge relies heavily on its *absorptive capacity*: "the ability of a firm to recognize the value of new, external information, assimilate it, and apply it to commercial ends" (Cohen & Levinthal, 1990, p. 128). Other issues like free riding and act of opportunism are also involved in this process. That is why establishing such routines demands partners to first develop 'partner-specific absorptive capacity'<sup>86</sup> to make information and know-how sharing feasible and easy as well as enhance socio-technical interactions (Dyer & Singh, 1998).

*Complementary resource endowments* is yet another indispensable part of RV and it refers to the "distinctive resources of alliance partners that collectively generate greater rents than the sum of those obtained from the individual endowments of each partner" (Dyer & Singh, pp. 666-667). These complementary resources are inseparable and indivisible in that neither firm in the partnership can obtain them in a secondary marketplace (Dyer & Singh, 1998). There are several challenges highlighted by Dyer and Singh (1998) with regards to this notion. For example, a firm must be capable to identify and evaluate partners with complementary resources. This relies on the extent to which a firm has access to precise and timely information on potential partners and their resources. Another challenge is system-culture compatibility of partners that enable them to coordinate actions (e.g., decision-makings).

Finally, *effective governance* plays a pivotal role in relational rent generation as it has a bearing on transaction costs and partners' willingness to get involved in value-creation initiatives. Governance structure, in other words, acts like a 'safeguard'

---

<sup>86</sup> *Partner-specific absorptive capacity* refers to the idea that "a firm has developed the ability to recognize and assimilate valuable knowledge from a particular alliance partner" (Dyer & Singh, p. 665).

which brings down the transaction costs thereby making it more efficient (Williamson, 1985). However, for safeguards to be economical (efficient) Dyer and Singh (1998) emphasize on *self-enforcing agreements* (as opposed to third-party enforcement of agreements). In sociology, these types of agreements are known as “trust/embeddedness”. Regardless what they are referred to in different disciplines, the self-enforcing agreements are categorized as formal and informal safeguards where the former encompasses a situation where participants are financial and investment hostages (e.g., Williamson, 1983); while the latter involves goodwill, trust and embeddedness (e.g., Powell, 1990).

Effective governance (including safeguards and agreements) is important as it may reduce opportunism. The informal self-enforcing type is of particular importance because it relies on personal trust relations or reputation as governance mechanism. Also, as cited in the literature, this type of governance seems to be the most effective and least costly means of safeguarding specialized investments and facilitating complex exchange (see: Hill, 1995; Sako, 1991; Uzzi, 1997). Therefore, based on Dyer and Singh (1998) for a firm to collect high rents, it would be beneficial to rely on informal self-enforcing safeguards as they bring the transaction costs to a lower level vis-à-vis formal contractual governance structures which are very costly to create, monitor and enforce if so required.

These four concepts elaborated above form the cornerstone of Relational View of the firm. Further, for a firm to sustain relational rents, Dyer and Singh (1998) emphasize interorganizational *asset interconnectedness, partner scarcity, resource indivisibility and institutional environment*.

For example, a specific highly specialized asset created as a result of the long-term inter-firm relationship based on mutual co-evolution of capabilities is useable in a

meaningful manner only as long as the relationship holds (e.g., Nissan-supplier conveyor belt; VISA brand name is collectively owned by participating banks). This suggests asset interconnectedness and resource indivisibility. In addition, the number of partners that have complementary strategic resources and relational capability is limited within an industry. Therefore, a first-mover advantage is a relevant issue. Lastly, institutional environment that fosters trust among trading partners may facilitate the creation of relational rents (North, 1990). For example, in Japan, the institutional environment fosters goodwill trust and cooperation which in turn helps partnering firms enjoy sustainability in generating relational rents (Sako, 1991).

Making inferences based on what has been mentioned so far, in general, the RV (Dyer & Singh, 1998) has several links with the idea of technological collaborations (strategic alliances), as well as open R&D and innovation process mainly because by expanding and complementing the resource-based view of the firm (see Lavie, 2006) it aptly maintains that a firm's critical resources may traverse firm boundaries and may be embedded in interfirm resources and routines. Therefore, it highlights *interfirm relationships* as an increasingly important *unit of analysis* for explaining above-average profits (Dyer & Singh, 1998) and emphasizes common benefits that alliance partners cannot generate independently (Lavie, 2006).

#### RV and strategic alliances

As stated earlier, RV of the firm is a promising perspective to adopt in order to better understand strategic alliances. Some authors (Gulati, 1995; Kale et al., 2000; Dyer & Nobeoka, 2000) have already emphasized the role of interfirm relationships as it is described in RV of the firm and especially in the process of governing collaborations. For example, Gulati (1995) stresses the role of trust and personal interaction in interorganizational relationships and shows how trust enables firms to reduce

dependence on equity structures to govern the relationships: suggesting that mutual trust between partners reduces the fear of opportunistic behavior.

Kale et al. (2000, p. 218) develops the notion of “relational capital”: “the level of mutual trust, respect, and friendship that arises out of close interaction at the individual level between alliance partners”. Relational capital can help firms successfully balance the acquisition of new capabilities (learning) with the protection of existing proprietary assets in alliance situations. Thus, relational capital minimizes the likelihood that an alliance partner will engage in opportunistic behavior to unilaterally absorb or steal information or know-how that is core or proprietary to its partners.

Finally, Dyer and Nobeoka (2000) build upon the original emphasis made in RV (Dyer & Singh, 1998) with regards to networks as central to explaining competitive advantage, and further use the case of Toyota to provide evidence that “a network can be more effective than a firm at the generation, transfer, and recombination of knowledge”. Although a network is a powerful means for success because it allows diverse and tacit knowledge to flow among its members, as in the case of Toyota, authors also emphasize the role of “coordinating principles” and nature of relationships as facilitators of know-how transfer and cooperation (Dyer & Nobeoka, 2000). For example, redundant ties are helpful in locating valuable knowledge; while strong ties act as proxy for trust/social capital to facilitate actual transfer of tacit knowledge.

Overall, RV distinguishes between legal contracts and self-enforcing agreements; proposing: “in general, self-enforcing mechanisms are more effective than third party enforcement mechanisms both at minimizing transaction costs and maximizing value-creation initiatives” (Dyer & Singh, 1998, p. 670). Thus, self-enforcing agreements



can possibly lead to alliances' having greater potential for relational rents. However, RV does not specify "*how*" alliances can create conditions to be able to enjoy self-enforcing agreements. Further, while talking about mechanisms (asset interconnectedness; partner scarcity; resource indivisibility; and institutional environment) that preserve relational rents, RV remains silent on the '*nature of underlying technology*' and '*common goals*' as mechanisms to preserve the relational rents.

Table 6.1. The Theories Potentially Relevant for Explaining Collaborative OSS R&D and Innovation Process

Name of theory (with authoritative reference)	Tenets of the theory (with representative quotes)	Empirical studies (within alliances literature)	Explanations for collaboration within alliances literature	Deficiencies/Gaps in theory
Resource-based view, RBV (Penrose, 1957; Wernerfelt, 1984; Barney, 1986, 1991)	Wernerfelt (1984, p. 172) defines resources as "those (tangible and intangible) assets which are tied semi-permanently to the firm." RBV's Assumptions: 1) Strategic resources are heterogeneously distributed across firms; 2) These differences are stable over time (i.e., resource immobility/stickiness). The RBV suggests a link between firm resources and its <i>sustained</i> competitive advantage; when resources have 4 key attributes: 1) valuable; 2) rare; 3) imperfectly imitable; and 4) there cannot be strategically equivalent substitutes for this resource that are valuable but neither rare or imperfectly imitable (Barney, 1991, pp.99; 101; 105-106). Thus, RBV views a firm as a collection of sticky and difficult-to imitate resources (Penrose, 1959; Barney, 1986) which can generate firm rents through protection and deployment of them (Mowery et al., 1996).	Eisenhardt and Schoonhoven (1996); Das and Teng (2000); RBV suggests the rationale for alliances is "value-creation potential of firm resources that are pooled together." Thus, resources attributes promise accentuated value-creation and thus facilitate alliance formation. Further, their general RBV of alliances, emphasizes significant role of partner resource alignment based on similarity and resource utilization and that alignment directly affects collective strengths and inter-firm conflicts in alliances.	Eisenhardt and Schoonhoven (1996): Further to resource obtainment, "strategic" and "social" explanations should be both considered to explain alliance formation. Firms in strong social positions or vulnerable strategic positions tend to form alliances.	<ul style="list-style-type: none"> <li>• In which appropriate context, the resources are more or less valuable (Priem &amp; Butler, 2001)</li> <li>• How the resources can be obtained and how do they generate sustainable rent? (Priem &amp; Butler, 2001)</li> <li>• When, where and how resources can be useful? (Miller &amp; Shamsie, 1996)</li> <li>• Through what processes particular resources provide competitive advantage? (Lawrence, 1997)</li> </ul>
Resource dependence theory, RDT (Pfeffer & Salancik, 1978)	"Interdependence is a consequence of the open-systems nature of organizations-the fact that organizations must transact with elements of the environment in order to obtain the resources necessary for survival" (p. 43). Thus, "to understand the behavior of an organization you must	Pfeffer and Nowak (1976); Stearns et al. (1987); Goes and Park (1997); Elg (2000); Yan and Gray	Early research (Pfeffer & Nowak, 1976) supports RDT's application to JVs finding that JVs are commonly formed between interdependent firms. Empirical evidence also supports the use of interorganizational relationships to reduce domestic and international environmental	<ul style="list-style-type: none"> <li>• Lack of integrating RDT with other theoretical perspectives to consider the dynamic nature of these dependencies and power as well as the multiplicity of interdependency (Hillman et al., 2009, p.1408).</li> <li>• More research examining</li> </ul>

	understand the context of that behavior—that is, the ecology of the organization” (p.1).	(1994, 2001)	complexity and gain resources (Elg, 2000; Goes & Park, 1997; Stearns et al., 1987). Yan and Gray (1994, 2001) examine the balance of power between international partners, finding that alliances occur when organizations are mutually dependent but the partner controlling more important resources retains strategic control. Inkpen and Beamish (1997) find that as one partner accumulates key resources from the other, the venture becomes less stable. Others (Levine & White, 1961; Aiken & Hage, 1968) mention their perspective that firms tend to enter partnerships if they perceive critical strategic interdependence with other firms in their environment; where Gulati (1998, p. 299) emphasize a condition where one organization has resources or capabilities beneficial but not possessed by the other. Richardson (1972), in a theoretical economic account, also proposed that the necessity for complementary resources is a key driver of interorganizational cooperation.	multiple forms of interdependencies may reveal of the conceptual perimeter of RDT (Casciaro & Piskorski, 2005)
	“RDT recognizes the influence of external factors on organizational behavior and, although constrained by their context, managers can act to reduce environmental uncertainty and dependence (Hillman et al., 2009, p. 1404).”			• More research considering the boundary conditions of RDT is needed (Pfeffer & Salancik, 2003: xxiv)
	[“R]esource dependence theory views interfirm governance as a strategic response to conditions of uncertainty and dependence...” (Heide, 1994, p. 72).]			
	[“The focus is on minimizing interorganizational dependencies and preserving the organization’s autonomy while recognizing that interorganizational relationships are necessary to acquire resources” (Gray & Wood 1991, p. 7).]			
Knowledge-based theory (KBT) of organizational capability (Grant & Baden-Fuller, 1995; Grant, 1996)	KBT rests upon 4 theory streams: a) competition as a dynamic process; b) RBV; c) organizational capabilities and competences; and d) organizational knowledge and learning. In KBT, “Knowledge” [emphasis on tacit/knowledge] is viewed as “the most strategically-significant resource of the firm (p. 375)” in dynamically competitive market environments. Central idea is: “the primary role of the firm, and the essence	Mowery et al. (1996); Kogut (1988); Dyer and Nobeoka (2000); Kale et al. (2000); Grant and Baden-Fuller (2004);	Various studies view strategic alliances as means for sharing knowledge (know-how, technology, and organizational capability) and “organizational learning” (Kogut, 1988). Hamel (1991) views collaboration (i.e., strategic alliances) as a means to internalize the other partner’s complex skills which are based on tacit knowledge. Alliances can short-circuit process of skill	• KBT identifies 3 characteristics of knowledge integration pertinent to competitive advantage: a) efficiency of integration; b) scope of integration; and c) flexibility of integration.
				• KBT emphasizes “flexibility of integration” in order to sustain competitive advantage through 2 dimensions: a) extending

	<p>of organizational capability, is the integration of [specialized] knowledge [of many individuals] (p. 375)." Knowledge plays a significant role in organizational capability. KBT's assumptions about success: 1) "under dynamic competition, superior profitability is likely to be associated with resource and capability-based advantages than with positioning advantages resulting from market and segment selection and competitive positions based upon some form of 'generic strategy'; and 2) "such resource and capability-based advantages are likely to derive from superior access to and integration of specialized knowledge (p. 376)."</p>	<p>acquisition. Further, capacity to learn (i.e., absorptive capacity) as in case of internalizing partner's skill can create a bargaining power for either of the partners (Hamel, 1991).</p>	<p>existing capabilities to include new knowledge; b) reconfiguring existing knowledge into new types of capabilities.</p>
<p>Relational view, RV (Dyer, 1997; Dyer &amp; Singh, 1998)</p>	<p>RV: "a firm's critical resources may span firm boundaries and may be embedded in interfirm resources and routines" (Dyer &amp; Singh, 1998, p. 660). It emphasizes "relationship between firms" as "unit of analysis" to understand "competitive advantage." RV, further, highlights 4 potential sources of inter-organizational competitive advantage: 1) relation-</p>	<p>Mowery et al (1996, p. 89) find that "equity JVs appear to be more effective conduits for the transfer of complex capabilities than are contract-based alliances such as licensing agreements." They also emphasize the importance of absorptive capacity in acquisition of capabilities through alliances. Further, lower levels of transfer occur in unilateral contracts than in bilateral non-equity arrangements.</p> <p>Grant and Baden-Fuller (2004), distinguishing between knowledge generation and application, argue that alliances serve as a means to access knowledge, and that they can "overcome the limits of firms in encompassing highly differentiated knowledge integration processes, while offering efficiencies in knowledge utilization (p.77)."</p> <p>Gulati (1995) stresses the role of trust and personal interaction in inter-organizational relationships and shows how trust enables firms to reduce dependence on equity structures to govern the relationships: suggesting that mutual trust between partners reduces the fear of such opportunistic behavior.</p>	<ul style="list-style-type: none"> <li>• KBT highlights the role of networks (and "social networks": Liebeskind et al., 1996), and relational contracts as efficient conduits for knowledge integration, where there is lack of perfect correspondence between the firm's knowledge base and its set of products.</li> <li>• However, there is lack of explicit arguments over "how" relationships among actors (firms, individuals) facilitate knowledge integration.</li> <li>• It also does not explicate the role of necessary capabilities such as absorptive capacity in integration process.</li> <li>• KBT also does not pay attention to other types of knowledge such as "know-who" or "know-why" as facilitators of knowledge integration process.</li> <li>• RV distinguishes between legal contracts and self-enforcing agreements; proposing: "in general, self-enforcing mechanisms are more effective than third party enforcement mechanisms both at minimizing transaction costs and maximizing value-creation</li> </ul>

specific assets; 2) knowledge sharing routines; 3) complementary resources and capabilities; and 4) effective governance.	<p>Kale et al. (2000) develops the notion of "relational capital: "the level of mutual trust, respect, and friendship that arises out of close interaction at the individual level between alliance partners (p. 218)." Relational capital can help firms successfully balance the acquisition of new capabilities (learning of know-how) with the protection of existing proprietary assets in alliance situations. Thus, relational capital minimizes the likelihood that an alliance partner will engage in opportunistic behavior to unilaterally absorb or steal information or know-how that is core or proprietary to its partners.</p> <ul style="list-style-type: none"> <li>• initiatives" (Dyer &amp; Singh, 1998, p. 670). Thus, self-enforcing agreements can lead to alliances' having greater potential for relational rents.</li> <li>• However, RV does not specify "how" alliances can create conditions to be able to enjoy self-enforcing agreements.</li> <li>• Further, while talking about mechanisms (asset inter-connectedness; partner scarcity; resource indivisibility; and institutional environment) that preserve relational rents, RV remains silent on the 'nature of underlying technology' and 'common goals' as mechanisms to preserve the relational rents.</li> </ul>
<p>Transaction cost economics, TCE (Coase, 1937; Williams on, 1975, 1981, 1985, 1998)</p>	<p>TCE regards "transaction as the basic unit of analysis" (Williamson, 1981, p. 548). There are 3 major transaction costs: 1) search and information; 2) bargaining and decision costs; and 3) policing and enforcing costs (Coase, 1937; Varian, 2002). TCT's assumptions in regards to the economic behavior of agents are: 1) agents are rationally bounded; accordingly all complex contracts become unavoidably incomplete (Williamson, 1998); 2) they are opportunistic which implies that firms are formed for "attenuating the ex post hazards of opportunism by</p> <p>Hennart (1988); Kogut (1988); Pisano (1989); Williamson (1991); Gulati (1995)</p> <p>Literature on strategic alliances has focused extensively on 'partner opportunism' examined mainly from theoretical viewpoint of TCE (Hennart, 1988; Kogut, 1988; Pisano, 1989; Williamson, 1991). In fact, a firm's concern about their partners acting opportunistically is likely to lead to higher transaction costs. To ward off this potential weakness, research has suggested firms to adopt appropriate contractual agreements or governance structures to address these concerns. For example, Kogut (1988, p. 321) raises this issue of "<i>mutual hostage positions</i> through joint commitment of financial or real</p> <ul style="list-style-type: none"> <li>• Zajac and Olsen (1993) highlight TCE's weaknesses as: its "single-party analysis [as opposed to multi-firm orientation] of cost minimization" (p.132); and that TCE overemphasizes structural aspects (ex-ante/ex-post dichotomy vs. process). Thereby, the former leads to the negligence of the inter-dependence between exchange partners; while the latter ignores "processual aspects" (p.135).</li> <li>• Ozman (2009, p.43) classify</li> </ul>

ex-ante choice of governance” (Williamson 1998, p. 31). TCE describes the firm as a governance structure (Williamson, 2005; Williamson, 1998). Thus, “economic institutions of capitalism have the main purpose and effect of economizing on transaction costs (Williamson, 1985, p. 17).” Thus, according to Williamson (1975), the organization of economic activity is a decision between markets or hierarchy. For example, vertical integration can be the efficient solution to a transaction cost minimization problem, where the costs of market exchange compare unfavourably with the costs of controlling production hierarchically through ownership. Williamson (1985) extends this logic to also encompass inter-firm relationships falling between markets and hierarchies.

assets” as this leads to “superior alignment of incentives”, and stabilizing “the agreement on the division of profits or costs”.

Pisano (1989)’s analysis of the motives for using partial equity investments (i.e., JVs) in collaborative relationships

demonstrates that the use of equity linkages (as in biotechnology collaborations) is related to the potential transactional problems associated with transaction-specific knowledge, uncertainty, and small numbers bargaining conditions. Thus, due to partners’ ex ante commitments to an equity alliance their concern for their investments reduces the possibility of opportunistic behavior over the course of the alliance (Pisano, 1989).

Finally, Gulati (1995) challenges the perspective offered by TCE concerning alliances as it uses “a singular emphasis on transaction costs” due to partner opportunism and advocates the use of contractual agreements or equity to resolve the issue. Such approach therefore gravely ignores the role of inter-firm trust and the evolution of inter-partner relationships (Gulati, 1995)

criticisms to TCE into three: 1) notion of opportunism which can be obviated based on mutual trust between partners (forming a network); 2) the need for a more dynamic approach to understand networks in environments of rapid change (network governance issues go beyond the dichotomy of firm vs. market); and 3) the lack of social processes in TCE.

*Source: Compiled by the author.*

## 6.5 Theoretical consolidation and possible explanations in the context of OSSTC

### 6.5.1 Actors, interdependence, and resources-based views

The ‘interdependence’ of ‘independent key actors’ emerged in several cases I studied, and it was explained from empirical perspective in detail in Chapter 5. Obviously, OSS firms, enterprise clients, hobbyists, and community of OSS project engage in technological collaborations, having heterogeneous incentives. Irrespective of the differences which exist about their motivations, a closer look at their incentives show that collaborating partners are in quest for accessing (strategic) ‘resources’. Thus, *accessing knowledge (tacit know-how), enhancing technological capabilities, learning new competences and creating the state-of-the art knowledge* (Case of FFmpeg<sup>87</sup> technology), and *obtaining efficiency* (time and cost of development) are among key incentives shared by different groups of independent actors.

TCE’s (Coase, 1937; Williamson, 1975) theoretical perspective – which undermines or even “excludes cooperation” (Niosi, 1995, p. 8) – has adopted singular focus in viewing economic organizations through the dichotomy of either markets or hierarchies. Therefore, it cannot explain the underlying reasons about voluntary and tight collaboration among firms and network of individuals within OSS RDIP. More specifically, TCE, by overemphasizing the role of hazards associated with opportunism and their negative influence on increasing transaction costs, has deliberately ignored other plausible explanations (e.g., value associated with learning) which induce collaborative behavior among actors. Lastly, OSS technology is

---

<sup>87</sup> For example, the team of product engineering of Savoir-faire Linux tried to address three FFmpeg’s technological challenges in order to be able to further shape and adapt the TR-03 standards for broadcasters. This R&D effort required them to look at FFmpeg’s internals in the hope of getting a working TR-03/SDI pipeline processing up to a several HD streams on a contemporary server while benefiting from Ffmpeg’s easily available lower definition derivatives from the same video stream. About FFmpeg: it is the leading multimedia framework, able to decode, encode, transcode, mux, demux, stream, filter and play pretty much anything that humans and machines have created. It is licensed under LGPL 2.1+ and GPL 2+. Available at: <https://ffmpeg.org/about.html>.



regularly built upon the collective power of community of developers. 'Diversity of knowledge' and 'continuous development and improvement' are two key features of its perceived technological superiority over closed software technology. Therefore, even if acquiring an OSS project is possible, and in some cases it is also cost efficient, keeping the project open within the commons makes sense, for as long as common benefits associated with OSS development methodology are sought for. Thus, the core features of OSS technology development methodology, at heart of which lies technological collaboration, cannot be accounted for by a mere dichotomy perspective of market vs. hierarchy; thereby, highlighting the serious deficiencies of TCE with regards to explaining the OSS technological and strategic cooperation. Moving beyond TCE, however, other four theories (RBV; RDP; KBT; RV) seem to be more appropriate to explain OSSTC.

RBV (Penrose, 1959; Wernerfelt, 1984; Barney, 1986, 1991) puts strategic resources and their role under spotlight. RVB, therefore, provides the rationale for formation of strategic alliances as these contract-based collaborations enable firms to access and/or acquire what they cannot otherwise obtain. RBV can also shed light, to a great extent, on OSSTC. Yet, it does not fully capture all aspects of collaboration among actors.

On a more detailed note, let us consider the two assumptions of RBV in the context of OSS industry. For instance, strategic resources (i.e., OSS-related know-how, trained and experienced human capital and technological capabilities) are indeed heterogeneously distributed across firms; and in case of OSS projects across individuals and platforms as well. However, the assumption which notes that the differences among firms (and individuals, in case of OSS) are stable over time does not very well reflect the nature of OSS development process. This is mainly because OSS RDIP is about developing open and collaborative technology platforms which are continuously evolving and creating new knowledge. The strategic knowledge



resources owned by a particular firm or a particular group of individuals alone cannot per se lead to sustained competitive advantage. This is mainly because *OSS strives within an open ecosystem or through OCTP in which there is no 'stable' knowledge equilibrium*. This issue of having evolving and multiple knowledge equilibriums makes OSS a unique case. In fact, a firm can gain sustained competitive advantage through using proprietary-licensed customized strategic OS features and modules in its manufacturing process; nonetheless, since the top layer (or strategic part of technology) relies on a supporting layer which is shared in the commons or the platform and which cannot be owned perfectly by a single firm, the competitive advantage of the firm cannot be sustained unless the supporting layer is sustained too. This *technological interdependency* within OCTP, thus, links and relates all independent actors together in a particular fashion – which can be explained partly through RDP and partly through RV of the firm. More specifically, these theories are complementary because RBV has remained silent with regards to ‘*how*’ key resources can be obtained and *under what circumstances* and through *what processes* they can generate rents (see Priem & Buttler, 2001).

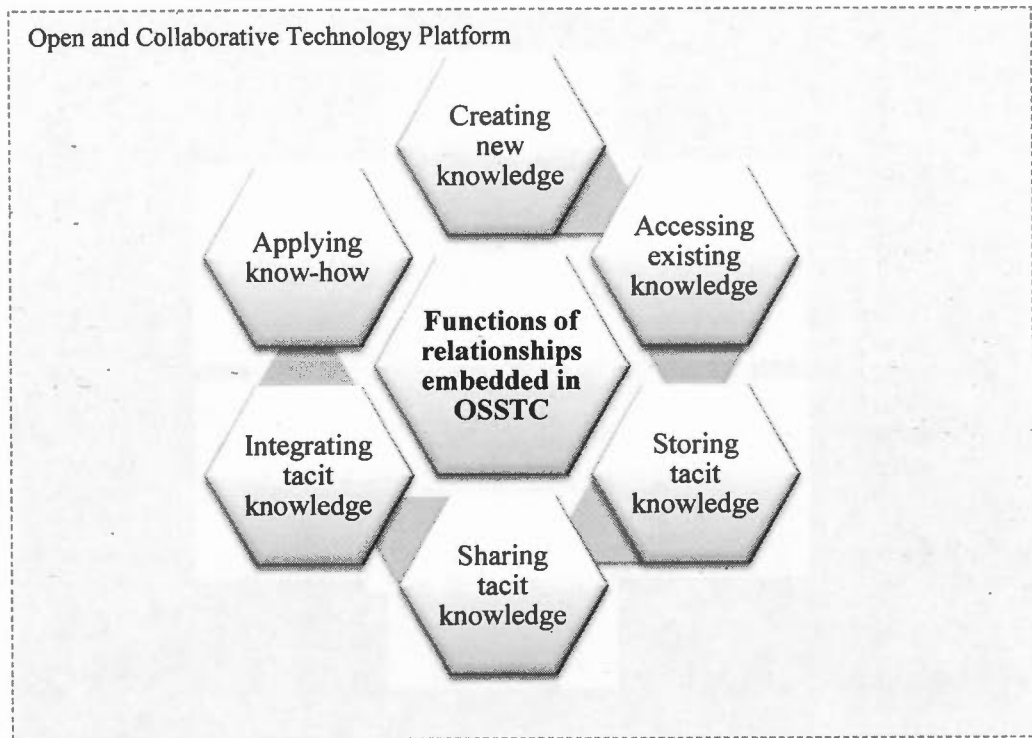
According to RDP (Pfeffer & Salancik, 1978), interfirm relationships are formed as a strategic response through which managers can control their dependencies and reduce the uncertainty associated with them. Such perspective appositely can be used to explain why OSS firms, enterprise clients, core development teams of communities and hobbyists are together viewed as an interdependent system and schematically represented through the Borromean Links (see Chapter 5, Section 5.2). As explained earlier in Chapter 5, each actor group brings in a different type of know-how, and technological capabilities which complement those of the other actor groups and ensemble create the superior technological synergy. This complementarity perspective of resources which have been also mentioned in the literature of alliances as key driver of interorganizational cooperation (e.g., Levine & White, 1961; Aiken

& Hage, 1968; Richardson, 1972; Gulati, 1998; Das & Teng, 2000) clearly explains the interdependence and interconnectedness among actors within OSS ecosystem and more particularly within OSS shared R&D and innovation processes.

However, much of OSS knowledge resources and technological capabilities are tacit (i.e., OSS know-how or in industry terminology: “OS expertise”); dispersed among and residing within individuals’ heads. KBT has already emphasized “knowledge” as the most important strategic resource of the firm, and that the main function of firm is to integrate the dispersed, specialized knowledge (Grant, 1996; Grant & Baden-Fuller, 2004). It has also discussed the specific role of organizational routines (as opposed to direction) as a mechanism for tacit knowledge transfer and integration (Grant, 1996). It also highlights the continual renewal of gained competitive advantage through: a) extending the existing capabilities to encompass additional knowledge; and b) reconfiguring existing knowledge into new capability types (Grant, 1996).

If we remain focused on tacit knowledge or know-how as the key strategic resource to build, maintain, and re-innovate OSS tools and modules; then, both RBV’s and KBT’s arguments to obtain and integrate this strategic resource to remain relevant to competitive arena are plausible. However, the key questions remains as “*how*” to obtain them and “*under what circumstances*” obtaining knowledge resources are most efficient and effective (valuable). In OSS industry, “relationships” are considered as the ‘central conduit’ for *creating, accessing, storing, sharing, integrating, and applying* know-how embedded in OCTP. These six functions (Figure 6.1), therefore, shift our attention to RV (Dyer & Singh, 1998) to further explain OSSTC.

Figure 6.1. The Six Functions of Relationships Embedded in OSSTC



Source: The author.

### 6.5.2 Technological collaboration and relational view

Strategic/technological alliances are mainly “*long-term contractual agreements* between two or more *enterprises* aiming at the development of new or improved product or process technologies” (Niosi, 1995, p. 3, emphasis added). OSS collaborative RDIP (as a proxy for strategic and technological collaborations within OSS industry) bear certain traits that put them in an awkward position when compared to strategic/technological alliances.

On the one hand, just like strategic alliances, OSS projects can be long-term and based on written contractual agreements orchestrated through foundations and

associations (e.g., Eclipse Foundation). On the other hand, many of them are based on “relational contracts” (Macneil, 1974, 1980) formed between individuals or between firms and individuals; and, they can even be short-term. Even in the context of formal OSS collaboration agreements, as emphasized by almost all interviewees, a huge bulk of collaborations takes place through relationships governed by trust and familiarity. Thus, R&D and innovation efforts within OSS projects can, to a great extent, rely on relational exchanges.

Opposite to market exchanges or so-called watertight contract-based exchanges such as JVs, relational exchanges take place on the basis of a historical and social context, such as trust. Heide (1994, p. 74) asserts that “relational exchange . . . accounts explicitly for the historical and social context in which transactions take place and views enforcement of obligations as following from the mutuality of interest that exists between a set of parties . . .”. Furthermore, OSSTC resemble greatly the interorganizational “rent-generating process” which is emphasized through the “RV” (Dyer & Singh, 1998). These interorganizational relationships which are sources of competitive advantage in that they can generate rents are viewed as idiosyncratic ties that are typically different from arm’s length relationships based on contracts (e.g., case of strategic alliances).

Although RV of the firm originally spotlights “firms” and their “inter-organizational collaboration” thorough “inter-relationships”, I can extend the theory to include a wider range of innovators such as OSS developers as individual actors (both professionals, and hobbyists), OSS foundations and associations, expert teams, and so on. The core issue here is that collaborating entities can generate relational rents through “relation-specific assets”, “knowledge-sharing routines”, “complementary resource endowments”, and “effective governance” (Dyer & Singh, 1998).

I must emphasize, applying the relational rent to OSS industry does not necessarily reflect the original sense in the RV of Dyer and Singh (1998). RV views relational rent as “a supernormal profit jointly generated in an exchange relationship that cannot be generated by either firm in isolation and can only be created through the joint idiosyncratic contributions of the specific alliance partners” (Dyer & Singh, 1998, p. 662). Within OSS context, although actors may not necessarily gain “supernormal profit” or ‘above average profit’ in the sense it has been described within strategic management literature (March, 1991; Dyer & Singh, 1998); because of their investment in building interrelationships, they do benefit greatly from their collaborations. Chapter 5 elaborated in detail on each actor group’s motivations and incentives to engage in interrelationships and enumerated the benefits which will accrue to them. In short, OSS technological collaborations lead to: a) reducing the cost of software R&D; b) shortening the time to market; c) enhancing in-house technological capabilities and absorptive capacity; d) building new competencies through learning and accessing new know-how; e) improving branding and marketing efforts. Having viewed these benefits as ‘relational rents’ that participants can enjoy as result of their collaborations, I can move forward and examine the four tenets of RV.

Within OSS technological collaborations, the “relation-specific assets” are mainly “human assets” (Dyer & Singh, 1998) and their specialized know-how or tacit knowledge. This specialized know-how is generated, accumulated, and stored through long-term relationships working on different projects. When OSS developers engage in intensive and iterative interactions to develop, and improve the quality of a feature, and get it integrated into the core body of codes, they exploit “knowledge sharing routines” (Dyer & Singh, 1998) between other human assets in OSS communities. As these human assets have already been part of the OSS communities through prior experiences, they have a high level of “absorptive capacity” (Cohen &

Levinthal, 1990) which allows them to establish regular pattern of interfirm interactions which allow transfer, recombination or creation of specialized knowledge (Grant, 1996).

Furthermore, the human asset which exists within each actor group (OSS firm, clients, communities) is a “complementary resource endowment” (Dyer & Singh, 1998). Each actor group brings in a set of particular skills and technological perspective, feedback, and tacit knowledge to the technological collaboration. Together, not only they complement one another, but also can generate a ‘*synergistic value*’ that cannot be generated and sustained over time individually.

More particularly, accessing the tacit knowledge, learning new competences and enhancing technological capabilities which are among key motivations of getting involved in OSS collaborative RDIP are at the disposal of those who have heavily invested in the process. Therefore, the hazard of free-riding is, to a great extent, reduced. For example, a free-rider cannot enjoy inducing a behavior in another party through relational contract because S/he has not built the relational trust through prior intensive collaborations. Indeed, within OSSTC, self-enforcing agreements provide effective governance which greatly reduces the transaction costs and eases the transfer of know-how among project members.

To conclude, RV complements RBV, KBV, and RDP. Yet, RV does not fully explain OSSTC because it is focused on firms and super normal profits through generating rents. Furthermore, my observations and interview data add some new dimensions to both empirical and theoretical perspectives on classic strategic/technological alliances. These new dimensions are used to differentiate this research from the prior efforts and provide an ‘alternative perspective’ to better reflect the particularities of

OSSTC. Table 6.2 demonstrates the alternative perspective and sheds light on its points of departure from the existing theories.

#### 6.6 Open and collaborative technology platform as an alternative perspective

*"There is not, and never will be, a best theory. Theory is our chronologically inadequate attempt to come to terms with the infinite complexity of the real world. Our quest should be for improved theory, not best theory, and for theory that is relevant to the issues of our time."*

*Walsham (1997, p. 478)*

Despite previous insightful efforts (e.g., von Hippel & von Krogh, 2003; Weber, 2004; Kelty, 2008) to better understand the black box of OSS projects as a socio-economic phenomenon and to develop a theory for its technological collaborations; I believe OSSTC is still a phenomenon in search for a proper theory since it lies between formal collaborations (i.e., strategic alliances) and informal cooperation. Even it is more so because OSS phenomenon has been quite recently cited as the flagship of OI framework (e.g., Chesbrough & Appleyard, 2007) – a highly general and abstract phenomenon by itself in need of a sound theoretical underpinning (see Chapters 2). The case of OSS, as representative of OI, is so dramatic that sometimes the socio-economic and technological gravity of OSS out-expands its OI theoretical container. However, this shall not necessarily be viewed as a bad thing, because proponent of OI paradigm have themselves confessed to the thin theoretical background of OI; thereby viewing it mainly as a practice-driven concept—although it may have consumption for a layman who has not surveyed the literature on collaborative R&D and innovation management.

On a more detailed note, the key particularities of OSS technology makes it a unique case which cannot be adequately explained by a dominantly firm-centric approach such as OIP. In fact, modularity, evolutionary character, demonstrating traits of public goods (non-rivalry and non-exclusivity), depending on network externalities, and being highly process oriented developed through open and collaborative technology platforms are among the key features of OSS which extends the discussion well beyond an open vs. close dichotomy.

In this research, I have exploited a mix of participant observation, case study and grounded theory approaches and applied them to the theory development process. Specifically, GTA is a fitting methodology when the researcher intends to cast light onto the fine cracks that are persistent in the already existing theoretical concepts (see, for example, Hamel, 1991). It also enables a researcher to go beyond the conventional wisdom and extend the already existing theories by adding new conceptual blocks. The power of such research methodology lies mainly in taking a fresh perspective (see Locke, 1996) to the phenomenon under investigation and letting the systematic and empirical observations lead the researcher towards forming higher level abstract conceptual categories and stating their interrelationships.

By reviewing the major theories underpinning conventional (non-OSS) strategic/technological alliances, I managed to better position the emerging theoretical concepts of this research and evaluate their conceptual values. This has led to providing a finer-grained perspective on OSS technological collaborations within shared R&D and innovation processes. Table 6.2 presents the 'alternative perspective' resulting from this research and sheds light on its points of departure from the existing theories.



#### 6.6.1 OSS as a case of open and collaborative technology platform– OCTP

Parker, Van Alstyne, and Choudary (2016, p. 5) define a platform as “a business based on enabling value-creating interactions between external producers and consumers”. Thus, a platform is “an open, participative infrastructure for these interactions and sets governance conditions for them”; and, its main goal will be “to consummate matches among users and facilitate the exchange of goods, services, or social currency, thereby enabling value creation for all participants”. If a platform enables interactions among “multiple groups of surrounding consumers and ‘complementors’”, then it can be referred to as a “multi-sided platform” (MSP) (Boudreau & Hagiu, 2009, p. 163). In a nutshell, *value exchange* seems to form the centerpiece of any platforms be it from communication and networking industry (e.g., Facebook) or operating systems’ (e.g., iOS, Android, Microsoft Windows).

Others (Gawer & Cusumano, 2014) have taken a more precise approach in defining a platform and distinguishing between its two major types: *internal/company specific* vs. *external/industry-wide* platforms. The former refers to “a set of assets organized in a common structure from which a company can efficiently develop and produce a stream of derivative products”; and the later focuses on “products, services, or technologies that act as a foundation upon which external innovators, organized as an innovative business ecosystem, can develop their own complementary products, technologies, or services” (Gawer & Cusumano, 2014, p. 417).

The main attribute of platforms is their “network effect” (Katz & Shapiro, 1985, 1986) which basically refers to a situation where the number of users of a platform can either positively or negatively impact the value created for each user (see Parker et al., 2016). In case a platform generates a positive network effect, the more users join a platform, the more value is created for the whole network members– just like the case of Uber where more riders attract more drivers and vice-versa. “MSPs [and

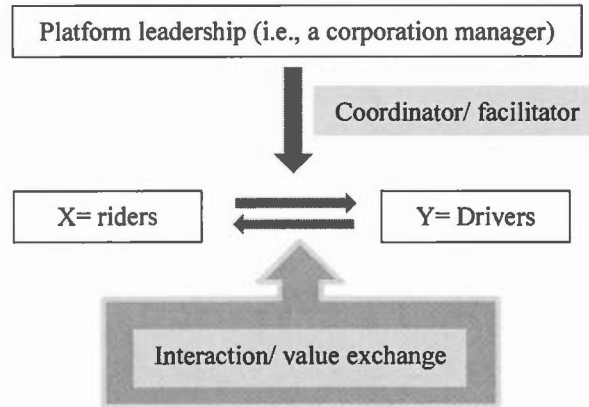
in general all platforms] are characterized by interactions and interdependence between their multiple sides” (Boudreau & Hagiu, 2009, p. 164). Parker et al. (2016) views positive network externalities as the “main source of value creation and competitive advantage”.

Although platforms share several core issues such as value exchange, network effects, and interdependence, not all behave similarly with regards to issues like platforms stability (reliability), long-term sustainability (viability) and overall collective success. More particularly, in case of “*technology-based platforms*” such as OSS platform where service and product offerings are inextricably intertwined, heterogeneity of members is at peak, and future of technology is both unforeseeable and debatable, the crux of the matter is more of ‘*what keeps the platform more stable, more viable, and more successful*’.

Earlier research (e.g., Parker et al., 2016) has mainly focused on company-specific platforms such as Uber. In case of ‘Uber’, the platform leadership deals with two main factors: riders (X) and drivers (Y) where driver's contribution is limited to offering a vacant seat in exchange of money (i.e., incentive) while rider's incentive is to fill in that vacancy by contributing money. Platform leadership, thus, coordinate and facilitate the relationship between riders and drivers (See Figure 6.2).

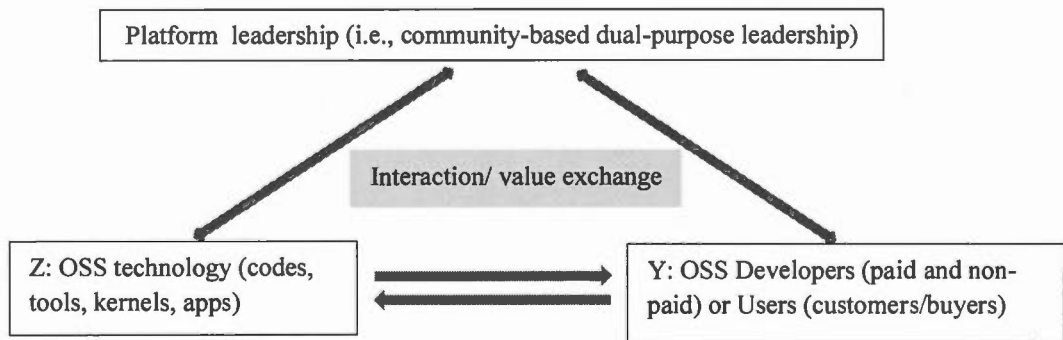
However, in case of OSS platform, it is more complex. First, there is another item to factor in, which is the ‘technology’ (Z) itself; one that is dynamic, that influences and can be influenced. Put simply, this third factor can bind platform participant or repel them if not managed properly. Moreover, the nature of leadership in an OCTP such as OSS platforms takes on different characteristics (See Figure 6.3).

Figure 6.2. A Model of Company-Specific Platform



Source: Author.

Figure 6.3. A Model of OSS OCTP



Source: Author.

Earlier literature on OI has extensively discussed the ever increasing phenomenon of openness in association with business models, strategic management, firm's innovation performance, and has even shed light on rudimentary yet fundamental questions in regards to degree of openness and its consequences. However, a genuine OSS technology platform can neither be classified in terms of strict proprietary forms of platforms such as the case of Uber, Google Apps, TopCoder, Amazon or facebook, nor can it be fully understood through the lens of OI where its core emphasis is on

how opening up the innovation process can benefit or hamper firm-level innovation performance. Indeed, OSS platform represents an open collaborative platform which can benefit from the doctrine of OI, but it deviates from the conventional thinking when it comes to one cornerstone of its strategic usefulness: i.e., long-term continuity and evolution. This central objective coupled with different nature and role of leadership leads us to view OSS technology platforms as an open and collaborative innovation platform where continuity is perceived as much critical as openness in the collaborative R&D and innovation process. Therefore, to characterize OCTP, I have identified seven dimensions and distinguish their nature from the existing literature.

#### 6.6.2 Unit of analysis

My theory of OSSTC embedded in OCTP is not only concerned with firms but also it is concerned with 'individuals' as building blocks of OCTP. It is also concerned with network of firms and individuals, role of communities of developers and all the interrelationships embedded in OSS communities. The distinction between firm-level and individual-level unit of analysis shifts the focus from considering drivers of competitive advantage from a corporate perspective to motivations and incentives of individuals who make significantly huge number of contributions to sustain the shared technology platform.

Furthermore, such theory tries to distinguish between "types of firms" involved in technological collaborations and platform development; namely, OSS firms, enterprise clients and for-profit proprietary software firms. Within the category of firms (more particularly about 'enterprise client'), one has to also make a more focused distinction as there are different types of enterprise clients (see Figure 5.11). Furthermore, by distinguishing between OSS firms, enterprise clients and other proprietary software firms, the theory can reflect the heterogeneous yet

complementary nature of resources that each actor group possesses; their impact on success and sustainability of OSS projects; and the challenges associated with their integration.

#### 6.6.3 Strategic resources and conduit for resources sharing

I think, a theory of OSSTC embedded in OCTP is essentially concerned with human assets or tacit knowledge (know-how). Thus, KBV (Grant, 1996) which squarely emphasizes the tacit specialized knowledge and role of firm as the integrator makes relevant discussions that can be applied to OSS technological collaborations. However, the new theory must also emphasize 'know-who' as a strategic resource. This is mainly because human assets carry the know-how and if they are not known by others and/or accessible then the know-how cannot be accessed, transferred, or learned (acquired).

Furthermore, the most 'efficient' and 'effective' conduit for accessing the know-how within OSS OCTP is the 'informal relationships'. The OSS engineers— who have worked on different OSS projects together— have over time developed a strong tie which is based on trust, a sense of familiarity, and certain level of learning capacity (absorptive capacity) which can facilitate the transfer of know-how in shorter time and at lower cost.

'Relationships' emerged as one of the main subcategories of the 'technological collaboration' category. It captures 'dyadic' and 'triadic' modes and sheds light on issues like two-way and tripartite knowledge sharing routines. The relationship typology provides an explanation for why certain projects are more efficient and effective in meeting end user demands and sustainability of OCTP than certain others. In my observations, in projects where clients (e.g., case of transport firm)

were more closely involved in developing and contributing to OCTP, were also more successful in creating network effects and influencing the technology trajectories. They could also more easily and accurately transfer know-how, and client's learning capacity increased greatly. Furthermore, the cost saving on client side was more than those clients who simply relied on OSS firms through dyadic relationships.

Thus, mainly contractual resources sharing routines may not work optimally within OSS projects embedded in OCTP. In OSS platforms most of the invaluable knowledge is tacit residing in individuals with ego and sense of OS or free software ideology. In this sense, RV (Dyer & Singh, 1998) which emphasizes interfirm relationships and knowledge sharing routines provides a pertinent explanation which fits why relationship building and the related capabilities are so important to build successful OSS technological collaborations. However, RV has little to say when it comes to the 'underlying logic' that derives actors to invest in building 'trust-based relationships' and invest in creating 'relation-specific assets' within OSS OCTP.

For instance, 'know-why' which has emerged as a category within the context for collaboration (see Chapter 5) highlights the genesis for starting to build a relationship. Within OSS context, educating, informing, and training private and governmental client enterprises as users of OSS technological tools that heavily depend on sustainability of OCTP are of paramount importance. It is helpful to show them *'why they should, at all, actively engage and collaborate in shared R&D and innovation processes'* and contribute to OCTP development. This sets the tone for top management to provide meaningful support for their IT departments by allocating them more resources (human resources, budget), 'expanded tolerance for risk' and 'decision making power'.

Finally, by taking a more encompassing RV towards understanding OSSTC, and by focusing on 'know-how', 'know-who', and 'know-why' we can also expand RBV (Penrose, 1959; Wernerfelt, 1984; Barney, 1986, 1991) on three fronts. First, it specifies the key strategic resources within OSS context; so that we should adopt a finer-grained resource perspective that emphasizes intangibles more than the tangibles. Second, the issue of "how" the strategic resources can be obtained is resolved through complementary perspective of RV. Third, 'know-why' sheds light on the point zero of embarking upon the relationship building journey in order to obtain resources and it also ensures the continuity of relationship maintaining and continuous strategic resources accessing.

#### 6.6.4 Governance

Within the theory of OSSTC embedded in OCTP, 'trust-based or self-enforcing' agreements can lead to effective governance. To ensure the trust-based agreements are optimally functioning and to further avoid annoying, ineffective, or opportunistic behaviors, 'dual-purpose leadership' complements the governance of OSSTC and the platform as a whole.

Throughout the literature on strategic alliances there has been a great emphasis on hazards of opportunistic behavior (e.g., Hennart, 1988; Kogut, 1988; Pisano, 1989; Williamson, 1991) and how equity controls (Hennart, 1988; Pisano, 1989) or "mutual hostage positions through joint commitment of financial or real assets" (Kogut, 1988, p. 321) can be used to reduce risks of opportunistic behavior; with some emphasis on social ties and role of trust (Gulati, 1995; Kale et al., 2000). However, the 'nature' of OSS technology, and the 'nature' of benefits firms and individuals receive through contributing to OSS OCTP are different from those discussed under mainstream alliances literature.

OSS technology is characterized through: 1) modularity, 2) interoperability needs, 3) evolutionary character, and 4) network or platform externalities (Banik & Taherizadeh, 2014). Modularity refers to the fact that software is created efficiently by combining and integrating components that have already been created. However, in order for modularity to be exploited, it requires interoperability or compatibility, meaning that the components of a system must function smoothly when they are assembled together. Compatibility is a primary concern for software designers, since software provides a service that often cannot be disrupted. Development of future code must always therefore be forward looking (anticipating future needs) as well as retrospective (ensuring that older systems can continue to work). Evolutionary character means that the software that is developed is always subject to being modified and improved. As an example, the average time before a new version of a mainstream Linux desktop operating system, is introduced is now just six months. This means that software designers must continually control what gets updated and weigh the pros and cons of introducing new changes that could compromise the stability and user base. Finally, the network externality refers to the benefits of a greater number of users sharing and developing the same software. Common standards and lower learning and costs are typical network externalities. Increases in the user base allows the OSS product to be more attractive to users (as it often results in greater support and development) developers and even OSS firms because it enables one to leverage benefits of standardization.

These four characteristics capture the nature of OSS technology. Obviously, actors that depend on OSS OCTP have great incentives to avoid opportunistic behavior and work together even in the absence of third-party enforcements.

In addition, each actor group; be it OSS firm, enterprise client, or community of developers can serve their self-oriented needs and benefit from their collaboration in



the underlying platform development. For example, OSS firms enjoy building and boosting their own in-house R&D technological capabilities. Enterprise clients develop and maintain their desired feature through developer community (i.e., commons). Core developer teams or community in general, attract financial resources and end-user perspectives by remaining in relationship with enterprise clients and OSS firms. These private benefits are therefore reasons to avoid opportunistic behavior and abide by the self-governing mechanisms.

On top of these issues, OSS OCTP benefits more from triadic or network-level relationships. I specifically use the term 'triadic' to make sure it includes enterprise clients on par with OSS firms, and other community developers. 'Network' level may be a confusing term here because it is all-encompassing and includes a lot of things without discriminating between the type and characteristics of the relationships.

Triadic relationships have high potentials for resources transfer and more specifically know-how. However, they are more prone to conflicts and disagreements which could be personal or technological and could eventually lead to project forking (as discussed earlier in Chapter 5). Thus, to ensure effective governance, a 'dual-purpose leadership' is needed to act as a mortar and glue the actors together, irrespective of their differences in wants, needs and technological capability levels. I have already explicated the conceptual building blocks of this concept, yet here I reemphasize the significant role of dual-purpose leadership in ensuring that: a) the relationships remain steadfast as conduit for resource sharing through time of collaboration; and b) the technological collaborations are managed, governed, and led smoothly, efficiently, and effectively. Dual-purpose leadership, as it means in this context, thus, is a new concept in my theory and it complements the RV (Dyer & Singh, 1998) in that it explains how alliances, here OSS technological collaborations, as well as OCTP can create conditions to be able to enjoy self-enforcing agreements with less risk of dissolution and/or opportunism.

Table 6.2. An Alternative Perspective of OSSTC Based on Observations and Emerging Themes from OSS Industry

Dimensions (Categories)	Existing theoretical perspective				Alternative (and complementary) theoretical perspective
	RBV	RDP	KBV	RV	
Unit of analysis	Firm	Firm	Firm	Pair or network of firms	<ul style="list-style-type: none"> <li>Technology platform,</li> <li>Pair or network of firms; Pair or network of firms and individuals; Network of different communities of developers and their inter-relationships,</li> <li>Including differences in types of organizations: OSS firms vs. enterprise clients vs. proprietary software firms</li> <li>OSS enterprise client typology</li> </ul>
Key strategic resources	Tangible/ intangible	Tangible/ intangible	Mainly tacit Knowledge (know-how)	Tangible/ intangible	<ul style="list-style-type: none"> <li>Mainly intangibles: tacit knowledge/human asset,</li> <li>Know-who,</li> <li>Know-why.</li> </ul>
Resource sharing conduit	Mainly contractual	Mainly contractual	Mainly contractual	Interfirm relationships	<ul style="list-style-type: none"> <li>Interfirm and inter-individual relationships and social ties,</li> <li>Borromean Links concept.</li> </ul>
Governance	Mainly third-party enforcement	Mainly third-party enforcement	Mainly third-party enforcement	Emphasizes self-enforcement; mainly informal governance	<ul style="list-style-type: none"> <li>Mainly informal self-enforcing governance mechanism,</li> <li>Dual-purpose leadership.</li> </ul>
Collaboration logic	Access strategic resources to gain C.A.	Obtain necessary resources to manage dependence and uncertainty	Access and integrate specialized knowledge of individuals to gain C.A.	Create relations-specific assets to gain C.A.	<p><u>Private or inward-looking logic (corporate thinking)</u></p> <p>To serve private 'pecuniary' (e.g., reduce cost of software maintenance) and 'non-pecuniary' (e.g., enhance in-house R&amp;D technological capacity) incentives.</p> <p><u>Public or outward-looking logic (platform thinking)</u></p> <ul style="list-style-type: none"> <li>To "continuous" access technological know-how and human assets;</li> <li>To "continuous" create, share, improve, evolve the underlying codes (knowledge) and make</li> </ul>

<p>necessary adaption;</p> <ul style="list-style-type: none"> <li>To “<i>constantly</i>” ensure interoperability, sustainability, and success of the underlying platform.</li> </ul>				
Interpartner/collective learning (learning-by-contributing)	Asymmetries in interpartner learning can have negative impact on collaboration and bargaining power	Asymmetries in interpartner learning can have negative impact on collaboration and bargaining power	Asymmetries in interpartner learning can have negative impact on collaboration and bargaining power	<ul style="list-style-type: none"> <li>Interpartner learning has positive spillover effect in that: <ul style="list-style-type: none"> <li>it can improve the quality of underlying common OSS tools and platforms,</li> <li>it can lead to further knowledge sharing practices, creating <i>new</i> knowledge in form of code contributions, and innovations,</li> <li>it contributes to success, sustainability and inter-operability of underlying OSS tools and platforms,</li> </ul> </li> <li>Collective learning can strengthen the unity and cohesion of the OSS community,</li> <li>Learning by contributing: learning takes place by contributing to OSS development and OCTP,</li> <li>Routine and dynamic learning capability,</li> <li>Need for dynamic learning equilibriums justifies continuous learning,</li> <li>Collective learning and writing industry platform standards (e.g., FFmpeg) reduce uncertainty level of unknown technology paths.</li> </ul>
	Scarcely, valuable, inimitable and hard-to-replace resources	Ability to manage interdependencies and uncertainty	Organizational capability to integrate knowledge of individuals beyond firm boundaries	<p>Relation-specific investments; interfirm knowledge-sharing routines; complementary resource endowments, effective Gov.</p> <ul style="list-style-type: none"> <li>Interrelationships between and among individuals, network of individuals, firms and communities,</li> <li>Relationship-building capability,</li> <li>Relationship-maintaining capability,</li> <li>OSS-specific absorptive capacity.</li> </ul>

Note (1): C.A.: Competitive advantage

### 6.6.5 Collaboration logic

The theory that I am developing here is very much concerned with the ‘logic’ underlying the collaborations embedded in the OCTP. This is mainly because part of the collaboration takes place in the commons or public domain (i.e., OCTP) where it cannot be heavily guarded nor its fruits can be completely appropriated by the actors. It is also partly due to the fact that OSS technology is always an ‘unfinished business’; i.e., it is a technology in the state of flux: being added to and improved upon to make sure it functions smoothly in conjunction with other related tools and dependencies. Therefore, innovation participants must have sound logic to embark on allocating resources which could have been otherwise used, on a continuous basis.

In this research (see Chapter 5), the results shed light on each actor group’s different ‘private’ incentives which can be classified into ‘pecuniary’ and ‘non-pecuniary’ types. For example, an enterprise client’s logic to collaborate can be very well the case of getting the much needed new feature built and integrated into the community core version so as to save on further maintenance costs (i.e., pecuniary incentive). An OSS firm’s technological collaboration on a specific project may be purely to learn new technological capabilities interactively and enhance in-house R&D capacity (i.e., non-pecuniary incentive).

However, beyond private incentives which are the basis for the creation of the term ‘inward-looking logic’, there is ‘public or outward-looking logic’ which encompasses the greater good; i.e., sustainability of the technology platform. The nature of OSS technology, as discussed earlier, binds the actors together in that unless the underlying technology functions well and unless the so-called independent built-upon modules are up-to-date and in harmony with the core code, the users cannot enjoy the superior functionality of their OSS tools. For instance, in case where support for ‘Red Hat® enterprise Linux 5.0’ comes to an end on March 31, 2017, there will be a surge

of demand to upgrade to the latest version by organizations that depend on Red Hat® technology platform. The peril of remaining inactive is to accept high risk of exploiting vulnerabilities which include both security and functionality of information systems.

Thus, the alternative perspective on OSSTC highlights a different kind of logic for collaboration on a shared platform. It emphasizes the collaboration logic which underlines: a) continuous access to technological know-how and human assets; b) continuous creation, sharing, improving, evolving code bases and making necessary technological adaptations; and c) constantly ensuring interoperability among interconnected systemic modules. The ultimate goal is to make sure the target OSS project, its underlying tools, core code, its dependencies and related modules remain 'sustainable' throughout the time. The technological sustainability also encompasses the notion of 'technological success' which has emerged as a key category in this study.

In the strategic management literature, resource-based theories such as RBV, RDT, KBV have emphasized the logic behind accessing strategic resources as a means to gain sustained competitive edge, and of course supernormal profit gains. However, within the alternative perspective I am putting forward, the sustainability of OSS project (modules, tools, platforms) is a prerequisite to arrive at other tangible monetary gains. For example, the tire manufacturer that uses several OSS tools in order to be flexible with its information systems and manipulate operations at will (without being concerned about or tied to a proprietary software giant) shall gain competitive edge vis-à-vis other players in the industry because they are using a uniquely flexible system that can bring forward production and meet market demand sooner than the rivals. If winter arrives two weeks earlier, they do not need to negotiate to-be-made software changes on short notice with several proprietary

software vendors as they already have access to the software internal codes. However, they need to remain connected with the source code hosted by community because after all the underlying tool is in public domain.

Therefore, the alternative perspective provides logics which also demand a collaborative circumstance or condition for the resources to be effective and lead to competitive advantages.

#### 6.6.6 Learning

‘Interpartner or collective learning’ has a *positive spillover effect* in the theory of OSSTC embedded in OCTP. This is mainly because learning can lead to improving the quality of the OSS being collaboratively developed. Learning also positively influences knowledge sharing practices, code contributions, and innovations; and it positively contributes to success, sustainability and interoperability of OSS tools and platforms.

Furthermore, collective learning which is mainly ‘learning-by-contributing’ drives strength in unity and cohesion of the OSS community as it provides an invaluable incentive to those actively involved in the collaboration. Within OSS OCTP, learning is at disposal of those who are actually contributing to the OSS development process. Without getting his hands dirty through iterative and collaborative coding and mentorship process, an OSS developer cannot obtain the invaluable learning experiences. Therefore, the process of learning holds actors together as a learning unit which ensures cohesion throughout the process.

In addition, OSS technological collaborations reflect the core idea of “partner-specific absorptive capacity” which is conceived as key to learning and knowledge

assimilation within the RV (Dyer & Singh, 1998). Meaning that, to learn effectively, one must have the necessary OSS-related absorptive capacity. Some interviewees emphasized that the quality and nature of the question posed to the community reflects, to great extent, the depth of experience of the one behind the question. Therefore, top coders get attracted to well-formed intelligently-designed questions because they know they are getting engaged in an intellectual discussion.

However, the conventional perspective on learning within strategic management literature and particularly within the domain of strategic alliances is quite different. In fact, learning the much needed know-how and capabilities have been considered as one of the key reasons to justify an alliance (Kale et al., 2000). Within the literature of strategic management and alliances, some authors (Prahalad & Hamel, 1990; Hamel, 1991) have studied firms from “skilled-based view” which considers a firm as portfolio of ‘core competencies and value-creating disciplines’ (i.e., firm-specific skills) rather than a portfolio of product-market entities (Prahalad & Hamel, 1990). Having focused on ‘acquisitions of skills’ as the logic for alliance formation, authors have emphasized the role of ‘out-learning’ one’s partner in making the laggard learner dependent and redundant within the partnership (Prahalad & Hamel, 1990). Hamel (1991, p. 101) argues that in collaborations where learning in form of ‘internalization of know-how’ is the objective then the longevity and stability of the partnership may not be useful proxies for the collaborative success. Thus, the ‘race for learning’ has received considerable attention (Khanna et al., 1998; Das & Teng, 2000) emphasizing the possibility of opportunistic behavior of partners to outlearn one another and causing instability in partnership.

The above argument on existing asymmetries in interpartner learning and its negative impact on joint collaboration and bargaining power (see Hamel, 1991) is in line with RBV, KBV, and RDP in that each partner is in search for a strategic asset (e.g.,



know-how) to gain an edge and once the know-how or capability is internalized the collaboration can become instable as the target is hit.

However, OSS technological collaborations rely on technology platform which should be constantly evolving and therefore in need of *dynamic knowledge equilibriums*. Partners learn, learn further, unlearn, and learn again in the course of developing, improving, and maintaining the software tools and products. Thus, the learning process does not reach an end stage so that one partner feels independent from the other and leaves the relationship. There may be cases where actors get more involved in the collaboration and then distant themselves for a while in form of “*in-and-out relationships*”, yet in order to optimally benefit from the key OSS projects they need for their own operations, the partners need to remain connected and active within the OSS community. Consequently, the nature of technology and need for its sustainability make ‘learning’ a routine that pulls partners further together than drifting them away from their collaborative effort. In short, routine learning demands participants of OSS technological collaborations to develop a *dynamic learning capability* which extends through time and ensures sustainability of OSS OCTP.

#### 6.6.7 Primary sources of advantage

My theory of OSSTC embedded in OCTP perceives ‘primary sources of advantages’ differently from those mentioned in strategic alliances based on resources-based views (e.g., RBV, KBV). Traditionally, RBV emphasizes possession of the resources that are valuable, rare, imperfectly imitable, and hard to become substituted as critical to gain sustained competitive advantage (Barney, 1991).

RV (Dyer & Singh, 1998) shifts the spotlight on relation-specific (i.e., unique to interfirm relationships) assets, knowledge-sharing routines, complementary resource



endowments, and effective governance (i.e., mainly trust based) to provide a finer-grained explanation of role of resources and interfirm relationship in relation to gaining sustainable competitive edge. In this research, by looking deep into OSS technological collaborations, I can begin to shape a theory which complements these two perspectives and extend them by being more meticulous with sources of advantage.

Within OSS technological collaborations, individuals' social capital through strong and weak ties plays a significant role in enabling organizations and individuals to access human assets or know-how and build the opportunity to create relation-specific assets. Inkpen and Tsang (2005, p. 151) define social capital as: "the aggregate of resources embedded within, available through, and derived from the network of relationships possessed by an individual or organization". This is a powerful concept which emphasizes the paramount role of interrelationships as the bedrock for other important factors. For instance, knowledge-sharing routines, accessing and integrating complementary and specialized knowledge resources, collaborating without the presence of third-party enforcements, and creating as well as accessing dynamic multiple knowledge equilibriums to insert fresh know-how into the project, all in all rely on the interrelationships. Therefore, the 'relationship-building' and 'relationship-maintaining' capabilities are two significant complementary elements to interrelationships.

Prior literature has emphasized the role of interfirm relationships in form of alliances to access external knowledge resources (Gulati, 1998; Lorenzoni & Lipparini, 1999; Grant & Baden-Fuller, 2004)- a notion that has been viewed as firms' attempts to retain knowledge externally (e.g., see, Lichtenthaler & Lichtenthaler, 2009). Accessing external knowledge (Grant & Baden-Fuller, 2004) and "connective capacity" as the ability to establish links to other elements-making connections- to

facilitate knowledge access (Luhmann, 1995) have been re-emphasized in the more recent literature on knowledge management (Lichtenthaler & Lichtenthaler, 2009). Others (Kale & Singh, 2007) have also emphasized the role of “relational capability”<sup>88</sup> (Lorenzoni & Lipparini, 1999) in alliances’ learning process and viewed the concept as “the capacity of the organization to create, extend, or modify a firm’s resource base, augmented to include preferred access to the resources of its alliance partners” (Kale & Singh, 2007, p.996).

Yet, despite these efforts to highlight the role and importance of interfirm relationships in terms of “connective capacity” or “relational capability”; I still think that the two broader concepts needs to be broken down in order to be further clarified in the context of OSS technological collaborations; where most of the knowledge is tacit; and where the quality and existence of the individual-to-individual connection plays a significant role in facilitating generation and transfer of knowledge.

In the process of this research, the two notions of ‘relationship-building’ and ‘relationship-maintaining’ capabilities emerged as key ingredients which can complement the notion of “relational capability” (Lorenzoni & Lipparini, 1999) which is “capability to interact with other companies...accelerates the lead firm's knowledge access and transfer with relevant effects on company growth and innovativeness (Lorenzoni & Lipparini, 1999, p. 317).”

I define ‘relationship-building capability’ as: *The capability to get engaged with another actor (firm, individual, a team) in the process of code development and code or knowledge sharing based on mutual trust and reciprocity.* As the relationship is

---

<sup>88</sup> Relational capability is the “capability to interact with other companies...accelerates the lead firm's knowledge access and transfer with relevant effects on company growth and innovativeness (Lorenzoni & Lipparini, 1999, p. 317).”

built, it needs to be maintained to be effective. 'Effectiveness' here is central to benefiting from the relationship as it encompasses successful 'knowledge access', and 'knowledge transfer' between and among partners.

'Relationship-maintaining' capability refers to *the necessary actions actors must take to make sure the connection between actors is sustained throughout the shared R&D and innovation process*. For instance, remaining engaged in the iterative code developing process, sharing back the internal code developments with the community, sharing and integrating the specialized in-house know-how with other experts in the development process on a continuous basis are among cornerstones of relationship-maintaining capability.

In addition, 'OSS-specific absorptive capacity' is required to make sure each partner has enough technological competence and capability to understand the others and thus able to perform (to code) in harmony with the rest of the group. Originally, "absorptive capacity", acting like a sponge, refers to "the ability of a firm to recognize the value of new, external information, assimilate it, and apply it to commercial ends" (Cohen & Levinthal, 1990, p. 128). Within the context of OSS technological collaborations, a developer's prior OSS mentorship, personal and professional experiences<sup>89</sup>, past track records, cultural compatibility, and deep collaborations with OSS core developers, all in all, lead to building up a very specific OSS expertise which encompass: know-how, know-why, and know-who. This specific expertise is called here 'OSS-specific absorptive capacity' and I recognize its significant role in maintaining the relationships.

---

<sup>89</sup> By 'personal experience', here I mean the playing-around activities a developer takes part in in order to enjoy, have fun and simply learn a few things about OSS tools and technologies; while 'professional experience' captures the coding activities an OSS developer performs in capacity of a hired, salaried engineer within the context of his or her work responsibilities. The distinction is important because a professional experience comes with accountability towards superiors, clients and even professional community-based projects and this brings the discussions into the strategic OSS with real client market as opposed to for-fun projects.

Together, the relationship-building, -maintaining capabilities, and OSS-specific absorptive capacity form the primary sources of advantage within the theory of OSS technological collaborations in OSS OCTP. This is mainly because they ensure access and transfer of know-how as in tacit knowledge, and therefore success and sustainability of shared OSS R&D and innovation process and the platform as a whole. Lastly, once relationships are maintained, the element of continuity which is embedded in maintenance leads to 'different knowledge equilibriums' to co-exist. This in turn provides diversity of fresh knowledge which feeds OCTP, its growth, success, and survival.

## 6.7 Conclusion

I reviewed and critically analyzed five major theories (TCE, RBV, RDP, KBT, RV) which have been extensively used to approach, understand, and explain strategic and technological alliances. This analysis coupled with the particularities of OSS technology and the emerged themes from Chapter 5 can justify my claim that they cannot fully reflect and explain the nature of OSS technological alliances in the context of their shared R&D and innovation processes. It is especially so if we take a platform view of OSS development – i.e., OSS OCTP. Out of five theories, four; namely, RBV, RDP, KBT, and RV embody insightful concepts which can be applied to the context of OSS OCTP. I, therefore, extracted the relevant concepts and organized them around seven dimensions that I developed as a guiding torch. Such arrangement helped me clearly identify the areas where OSSTC is different from conventional strategic alliances (see Figure 6.2). Thus, the seven dimensions and the OSS-related conceptual developments and clarifications, together form an alternative perspective or a theory of OSS open and collaborative technology platforms.

## CHAPTER VII

### CONCLUSIONS AND RECOMMENDATIONS FOR FURTHER RESEARCH

#### 7.1 Introduction

This chapter provides the concluding remarks by discussing theoretical contributions, managerial implications, limitations, and the avenues for further research. First, this study makes five main theoretical contributions which fall at the crossroads of the research fields of OSS, OI, OBM, and strategic alliances based on empirical evidence gathered from OSS industry projects. Second, the present research also offers key managerial implications that can benefit practitioners not only in software and/or OS software industry, but also those from other different downstream industries (music and entertainment, transport, auto part manufacturing, financial and banking, to name a few) who consider either adoption and integration of OSS tools/solutions, or are thinking about expanding their reliance on OSS projects. Third, this study discusses three main limitations and finally it ends by recommending four ways to advance this line of research further.

#### 7.2 OSS technology: A useful empirical tool for developing OI and platform theories

My focus in this empirical research has been on the OSS technology which is inherently a *unique* technology (see Section 3.4). It is highly modular, being always in a state flux with uncertain technological trajectories and evolving industry standards. It demonstrates the traits of public goods and heavily depends on positive network externalities in order to have a sustainable RDIP. These particularities,

therefore, significantly influence the way OSSTC functions and how they are managed.

In the recent literature of distributed and collaborative RDIP (e.g., von Hippel & von Krogh, 2003), OSS has been tightly associated with the concept of OI (e.g., Chesbrough & Appleyard, 2007; Gruber & Henkel, 2006; Maxwell, 2006). This association has extended to the point that OSS has been perceived as “the poster child”, “a great exemplar” of OI concept, and/or a fertile ground to study and theorize about OI (Chesbrough & Appleyard, 2007; West & Gallagher, 2006; Morgen & Finnegan, 2010). On a more detailed note, OI – while emphasizing the adoption of OI strategy – encourages firms to open up their business models (i.e., OBM concept) in order to more effectively create and capture value from their OI-based initiatives (Chesbrough, 2007b). OSS – as popularized through the lens of OI in recent years – has been also studied as a showcase by those whose objective has been to look deep into the OSS firms’ processes and explain how OBM functions (e.g., Perr et al., 2010). More specifically, as OSS is gaining more and more momentum and is becoming an indispensable part of software industry, OI has become a more relevant framework to study how private enterprises have ventured into exploiting the opportunities offered by OSS industry (West & Gallagher, 2006).<sup>90</sup>

However, fundamentally speaking, OIP is criticized on three major fronts. It lacks *solid theoretical underpinning* (e.g., Lichtenthaler, 2011; van de Vrande et al., 2009) – an issue which is still valid (Vanhaverbeke & Cloodt, 2014). OI has been built upon fuzzy pillars so much so that it has been labelled for being “*old wine in new bottles*” (Trott & Hartmann, 2009) – meaning that, such a thing as closed innovation (which has been used to justify OI) never existed. Moreover, there are several research shortcomings, many uncharted research domains, and dearth of external validity of

---

<sup>90</sup> For example: Gassmann, (2006); Henkel (2006); West and Gallagher (2006); Dittrich and Duysters (2007); van de Vrande et al. (2009); Enkel et al. (2009); Gassmann et al. (2010).

the concept (see Chapter 2 for more details). Based on these caveats, one should take the usefulness and appropriateness of OI concept in studying OSSTC with a pinch of salt.

Moreover, based on my reviews of OI literature (see Chapter 2), I conclude that openness is the *essential* condition for adopting OIP, but it is *not the sufficient* condition to benefit from collaborative R&D. Issues like appropriate governance, managing network externalities, honing the right absorptive capacities, to name a few, are among the significant factors that must be discussed at a higher organizational level. In addition, the overemphasis of the existing literature in framing openness as *dominantly a firm-centric phenomenon* which is subject to firm-level managerial discussions (see, Piller & West, 2014) poses a limited conceptualization of openness and OI. This coupled with the dearth of understanding of “the interactive and reciprocal nature of “coupled innovation processes” (Chesbrough & Bogers, 2014, p. 19) becomes a more pressing issue specifically in case of OSSTC embedded in OSS R&D projects. OSSTCs include multi-sided platform-based interactions among a plethora of firms, community of developers and individuals who collectively develop software modules in the commons or the public domains. Thus, OI’s predominantly firm-centric approach does not adequately explain them.

Additionally, the review of the BM’s literature makes the case for claiming that it is not a well-equipped concept to deal with the openness, open strategy, and more importantly, OSS. BM, as it seems, is still very corporate-centric, focused on firm-level sustainable competitive advantage, customer segment and revenue focused, and the internal structure of the firm (See Chapter 2). Although the bulk of the literature tends to position the concept in the thinking of conventional strategic management, i.e., Porter’s five forces (1980a), recently scholars have acknowledged that BM

expands firm boundaries and that it is becoming more focused on value creation processes. Still, BM does not seem to have become attuned to OI or UI concepts where value creation and capture processes are not readily situated within organizational boundaries. It certainly is poorly suited to explain the core and nuances of OSSTCs.

Moreover, based on the discussions on OBM (See Chapter 2), we can conclude that, for BM to fit into OIP, it must adopt an open strategy and step outside the organizational boundaries. Put simply, BM needs to transform into OBM in order to become relevant to discussions on open and distributed innovation. The discussions on OBM, therefore, are more useful in the context of OSSTCs as they have begun to tap into ecosystem way of value creation, partner networks, platforms, and alliances. All these concepts, however, are subject to detailed investigation in order to reflect the particularities of OSS technologies and their development processes.

In addition, there is a shifting focus which targets the notion of 'value creation processes', its 'complexities', and the 'central role of customers' in designing 'customer-focused value creation processes' (see Zott et al., 2011; Teece, 2010, 2007; Morris et al., 2005; Chesbrough & Rosenbloom, 2002). Having realized that the processes of value creation and value capture are two distinct issues (although they can be interrelated), the former has received little attention compared to the latter (Nickerson et al., 2007). In addition, previous studies have mainly looked at the pivotal role of BM in technology commercialization process at the level of *individual firm* (see, for example, Chesbrough & Rosenbloom, 2002; Bjorkdahl, 2009; Calia et al., 2007). Therefore, they have ignored the *role of individuals, communities and networks* in creating value and commercializing the newly developed technologies- in short, paying little attention to importance of BM at industry level (Zott et al., 2011). Furthermore, the paucity of research in examining *the role of commercial clients* of



OSS solutions/products associated with issues such as success and sustainability of OSSTC and projects (which are embedded within the scope of OBM) have created a theoretical hollow in the intersection of OI, OSS, OBM research fields. Also, Piller and West (2014)'s latest attempts made to further theorize the concept of OI highlight more research gaps with regards to governance of collaboration process, collaboration infrastructures, and supporting capabilities. For instance, we need to understand whether or under what conditions OBM adopted by OSS firms can lead to positive network externalities and continuous value creation.

In consideration of these existing gaps, this dissertation makes several key theoretical contributions.

### 7.3 Theoretical contributions: Bridging OSS, OI, OBM, and strategic alliances concepts

The results of this research— as laid out in Chapters 5 and 6— advance the literature of OSS, OI, OBM, and strategic alliances in the following ways.

#### 7.3.1 Platform perspective vs. Firm-centric-view

I have characterized the particularities of OSS technology and studied OSSTC embedded in OSS RDIP adopting a lens of technology platforms. Therefore, this study breaks away from the mainstream OI approach – which is “a firm-centric paradigm that is primarily concerned with leveraging external knowledge to improve internal innovation and thus the firm’s economic performance” (Piller & West, 2014, p. 29). My perspective is also different from some of the earlier OSS researchers’ approach such as Weber (2004). For instance, Weber (2004, p. 8) clearly mentions:

“My starting point for explaining the open source process is the lens of political economy. I will situate the puzzle to start in modern concepts from political economy and then say more precisely why open source challenges some conventional theories about the organization of production, and how it affects and is affected by society.”

My theory introduces the phenomena of OSSTC and OSS RDIP as a technology platform-centric phenomenon, i.e., Open and Collaborative Technology Platform (OCTP). This view seems to better suit the key particularities of OSS and more comprehensively inform us about complexities involved in OSSTC and OSS shared RDIP. For instance, one of the reasons professionals and firms engage in collaborative development and maintenance of the OSS technology platforms is to possibly influence technology trajectories and shape the future industry standards to suit their products and services. If the platform remains closed to the public access, then they cannot create any network effects. Therefore platform openness ensures the potential for further adoption and creation of positive network effects; while continuous collaboration and contribution ensure sustainability and viability of the platform.

### 7.3.2 New knowledge co-development vs. Accessing and acquiring knowledge

OIP and theory of strategic alliances focus most of their attention on debating the strategic usefulness of *accessing*, or *acquiring* resources which lie beyond firm boundaries (See Chapters 2 and 6). Therefore, when OI or strategic alliances discusses the concepts of BM innovation, contract-based learning and resources sharing, collaborating on R&D projects, commercializing through second or third parties and gaining an edge in the incumbent industries, there is always a power asymmetry, a free-riding hazard, or simply an unfair condition due to incomplete

contracts underlying collaborations, where one partner may get the lion's share and win the collaboration race.

On the contrary, the results of this research (Chapter 5 and 6) show that in cases of OSSTC and OSS RDIP, there is another key incentive underlying collaborations; i.e., *'creation and co-development of new knowledge'* (i.e., software technology in form of codes, features, packaging methods, writing new industry standards, etc.) on an OCTP. Actors – either private firms as users or software developers or individuals or large public organizations– have logical incentives to remain connected and maintain their relationships– often in the absence of contracts– because if OCTP (i.e., the commons) grows and sustains, all participants will benefit from the underlying technologies. The incentives are logically justified not only based on firm-level or individual level benefits (as the case of past studies presented under Section 3.5.1, and Table 3.1) but also based on platform, network level or collective incentives. Such ecosystem-based view ignores the hazard of freeriding because software genuinely has the traits of public goods (i.e., non-rival, non-exclusive), so if more people join the OCTP and use it, it does not shrink in value. In fact, the more users join the OCTP, the higher will be the value of the network and platform – a phenomenon also known as network externalities. In addition, bilateral and multilateral contributions (see discussions on relationships in Chapter 5) ensure that the platform keeps developing and new technology standards are being added.

### 7.3.3 A 'detailed' analysis of the nature of OSS value creation processes

One of the main value-added and distinctive features of this research lies in its ability to identify key concepts related to OSSTC and OSS shared RDIP and explain them in extensive details. In fact, I delve into the *'nature of OSS value creation processes'*– as an important and indispensable part of OBM – and clearly identify actors and

explicate their key characteristics, incentives, roles, and how they collaborate (i.e., context for collaboration, relationship typology, code sharing complexities, and dual-purpose leadership) in order to illustrate why and how OSSTC and OSS RDIP can become successful and sustain long term. Others have also studied OSS collaborations in detail (e.g., Weber, 2004; Hang et al., 2005; Kelty, 2008; and many more, see Chapter 3), and significantly contributed to our understanding of the topic. However, concerning certain aspects associated with OSSTC and RDIP, they did not go to the so-called 'inner core' of the subject matter. One reason could be due to their adoption of only interview-based data collection (e.g., the case of Weber, 2004).

For instance, consider important concepts such as '*flexibility*' benefit provided by adoption of OSS-based information systems; '*learning*' as result of engaging in OSSTC and shared RDIP; '*know-who*' and '*know-why*' underlying technological collaborations; '*complexities associated with code sharing routines*'; '*dual-purpose leadership*' constructed ground up from minute conceptual building blocks gathered through a mix of qualitative methodology, including participant observation method. These details collectively enrich the developed theory (Figure 5.17. The Theoretical Model of OSS Technological Collaborations), and strengthen its explanatory power in terms of 'why', 'how', 'under what circumstances' (see, for more details, Whetten, 1989) and distinguish it from the earlier efforts. Several examples clearly highlight the contributions of this research to shed light on the *important nuances* of OSSTC.

For example, Weber (2004) highlights the issue of field testing; and, he mentions how the lack of rigor in testing can negatively impact OSS development. He further discusses and highlights the role of using *many diverse individuals* in order to generate patterns of use which are inherently unpredictable by the OSS developers. Nonetheless he does not go into more detail to treat diversity at a micro level. He does not identify and characterize enterprise users of OSS as a possible source of

knowledge feedbacks which can enhance and shorten the commercialization of OSS tools and solutions. However, based on the nuances that differentiate clients from one another according to their level of technological collaboration, I classify clients into four distinct groups (the conservatives, the luddites, the champions, and the reticent) and discuss their roles, capabilities, and incentives in detail (See, Chapter 5, Section 5.3.1, Part B, Figure 5.11).

Other examples include the notions of ‘flexibility’ (See, Table 5.2. Unpacking the Concept of Flexibility), and ‘meaningful cost saving benefit’ (Figure 5.10. Relationship between Clients’ Adoption of OSS Solutions and the Accrued Benefits). Past studies (e.g., Li et al., 2011) have identified these as “extrinsic motivation” but have not exactly studied their significance and nuances of their conceptual differences. Nor have they explained how these benefits can influence participants’ decisions to engage or not to engage in OSSTC and shared RDIP. The grounded theory approach (GTA) coupled with participant observation method provided me with a unique opportunity to understand, analyze and unpack these conceptual nuances and advance the OSS literature on these fronts.

Yet another example concerns the OI literature on the different approaches that firms adopt to manage their relationship with community of OSS developers (e.g., Dahlander & Magnusson, 2005, 2008). Dahlander and Magnusson (2005) propose a typology of three different approaches (*symbiotic*, *commensalistic*, and *parasitic*). However, they do not treat the *nature of OSS technology* as a significant factor that binds participants and provides an underlying logic for collaboration.

In this research, I identify inherent particularities of (OS) software and make a case for collective and sustainable development. The two concepts of ‘Water Purifier Model’ and ‘Borromean Links’ together illustrate the inescapable interdependencies

about software modules and their evolutionary character. These concepts further explain why firms and individuals may prefer a “symbiotic” relationship over “parasitic” one. Furthermore, the client typology (Figure 5.11) also discusses different cases based on their degree of collaboration (high vs. low) and type of connection (indirect vs. direct) with OSS communities. This type of detailed analysis provides a finer-grained understanding of different levels of collaboration and explains the underlying reasons— therefore, progressing the work of Dahlander and Magnusson (2005).

#### 7.3.4 Development of two detailed theories of OSSTC and their implications

The two theories that I have developed in this dissertations: 1) ‘An Inductive Theory of OSS Technological Collaborations’ (See Section 5.4 for details such as models, core propositions and hypotheses); and 2) ‘OCTP As an Alternative Perspective’ (See Section 6.6 for models and details on the seven dimensions of the theory) provide two complementary perspectives of OSSTC.

The first inductive theory— a bird’s-eye view of collaborations— identifies the key factors and explains their dependencies and how they influence (positively or negatively) the success and sustainability of OSSTC and OSS shared RDIP.

For instance, notions of success and sustainability are among the debatable and fuzzy concepts within OSS literature, and even more so as far as “success” of OSS projects is under spotlight (e.g., Asay, 2014; Sen et al., 2012; Subramaniam et al., 2009; Lee et al., 2009; Crowston et al., 2006; DeLone & McLean, 2003, 2002, 1992; Rai et al. 2002; Seddon et al. 1999; Seddon 1997; Raymond, 1999). By clarifying these concepts at different conceptual levels, and explaining how deeper layers of technological collaborations (e.g., dyadic and triadic relationships) and the

circumstances under which they can influence success and sustainability of OSS RDIP and OSSTC, this research provides a theoretical framework that can guide research for further empirical investigation. The four core propositions and the follow-up hypotheses try to explain and accentuate the roles of enterprise clients, dual-purpose leadership and tripartite relationships in contributing to success and sustainability of OSSTC.

On a more detailed note, the concepts of '*dual-purpose leadership*' and '*contextual factors*' in capacities of intervening and moderating constructs help us better explain how the relationship between OSSTC and outcome constructs (success and sustainability) are affected. It is the details as such which make this inductive theory a useful framework to study OSS phenomenon.

The second theory – which is partly inductive and partly elective– distinguishes OSSTC from the conventional strategic or technological alliances and clearly characterizes them based on 'seven key dimensions' (unit of analysis, key strategic resources, resources sharing conduit, governance, collaboration logic, interpartner or collective learning, and primary sources of advantage).

As a case in point, the literature of strategic or technological alliances views asymmetries in interpartner learning as negatively impacting the longevity of collaborations and shifting the locus of bargaining power from one partner to another. Thus, the 'race for learning' has received considerable attention (Khanna et al., 1998; Das & Teng, 2000) emphasizing the possibility of opportunistic behavior of partners to outlearn one another and causing instability in partnership. On the contrary, my theory – OCTP as an alternative perspective or the alternative perspective of OSSTC– views the notion of 'learning' differently and conceptualizes it in a manner that suits the OSS context, respecting the inherent particularities of OSS technology.

In fact, in my theory of OSSTC, interpartner learning has positive spillover effect in that it can improve the quality of underlying common OSS tools and platforms, lead to further knowledge sharing practices, creating new knowledge in form of code contributions, and innovations, and effectively contribute to success, sustainability and inter-operability of underlying OSS tools and platforms. Therefore, collective learning can strengthen the unity and cohesion of the OSS community. I also explain that learning takes place in form of 'learning by contributing'. This means that a free rider cannot learn much, a notion also highlighted by von Hippel and von Krogh (2003). Furthermore, I view learning as a 'routine learning' which is dynamic and requires a 'dynamic learning capability', therefore, the need for having dynamic learning equilibriums justifies continuous learning. Collective learning and writing industry platform standards (e.g., FFmpeg) also reduce uncertainty level of unknown technology paths.

### 7.3.5 Open and collaborative value creation platform as an example of OBM

The recent decade has witnessed an unprecedented number of attempts to further explore the BM black box from conceptual and empirical perspectives so as to improve its conceptual clarity as well as formulate empirical tools for managers to design BMs (Taran et al., 2015; Robins, 2013; Baden-Fuller & Haefliger, 2013; Casadesus-Masanell & Ricart, 2010; Teece, 2010; Baden-Fuller & Morgan, 2010). Among other individual publications (e.g., Zott & Amit, 2007), special issues<sup>91</sup> are evidence of the focused effort to converge scholarly attempts towards unifying the subject. While more definitions and perspectives on BMs would appear to be contributing to convergence of the definitions, in fact, their heterogeneity have allowed for divergence on several issues related to the topic. For example, in view of

---

<sup>91</sup> One can consult special issues such as Long Range Planning (2013, 2010), R&D Management Journal (2014), International Journal of Innovation Management (2013).



the shortening product life cycles, rising costs of R&D, shrinking margins due to stifling competition imposed by new (entrepreneurial) entrants firms have turned the spotlight on BM as a principal and standalone locus of innovation—a phenomenon which has been gaining momentum in the past decade, termed “business model innovation” (BMI).

However, our understanding of BM and BMI concepts is still limited. Moreover, based on the discussions on OBM (See Chapter 2), we can conclude that, for BM to fit into OIP, it must adopt an open strategy that functions beyond organizational boundaries. Put simply, BM needs to become open in order to be relevant to open and distributed innovation discussions. Thus, OSS and looking into OSSTCs provide a fertile research ground to develop some theoretical insights about internal workings of OBM (“how”) and “why” different players must actively engage in OSSTCs to create value long-term.

In Chapter 5, the sections on OSS technological collaborations and interdependence-independence shed light on *how* and *under what circumstances* (i.e., conditions and consequences) actors can co-create value and capture part of that value through boundary spanning collaborations. For instance, subcategories such as context for technological collaboration, code sharing modes, relationships typology, and dual-purpose leadership inform us of the details of the black box of collaborative and boundary spanning value creation processes within OSS projects.

Furthermore, the conceptual notion of Borromean Links (see Chapter 5, Section 5.3.3 Interdependence-Independence) provides insights on how the interdependence among actors is vital to creating a unique value for each participant that cannot be otherwise obtained (alone or independently), or through market transactions. This is mainly because within the collaborative OSS R&D and innovation process, *each actor plays*

*a unique and specific set of roles that is complementary to those of the others.* Thus, Borromean Links – which represents the twin concept of interdependence-independence (see Figure 6.16) – makes a theoretical contribution to understanding the nature of value creation process embedded in OSS business models and therefore OIB.

#### 7.4 Managerial implications

Table 7.1 selects and highlights the managerial or practice-oriented relevance of several concepts such as Borromean Links, Water Purifier Model, flexibility, (meaningful) cost saving, client typology, nature of client technological collaborations, context for and modes of collaborations, as well as success and sustainability.

Table 7.1. The Summary of the Key Managerial Implications

Theoretical concept	Managerial implications (recommendations)
Borromean Links Concept, and Water Purifier Model	<p data-bbox="555 466 1382 527">This concept can be used in designing the content of marketing and business intelligence documents in the following ways:</p> <ul style="list-style-type: none"> <li data-bbox="555 564 1382 660">a) To help the commercial clients understand the nature of OSS RDIP, and the underlying logic of iterative code sharing with communities,</li> <li data-bbox="555 662 1382 723">b) To inform the clients of their position as a key stakeholder in the R&amp;D process as opposed to a passive user,</li> <li data-bbox="555 725 1382 823">c) To motivate them to engage actively and invest in the iterative code sharing process through allocating more time, money, and intangible resources as well as permission to conduct share coding.</li> </ul> <p data-bbox="555 860 1382 1085">The Borromean Links further helps core developers and key administrators how to realize the vital role of enterprise clients in commercialization process of OSS. It can motivate them to change their perspective of clients from mere users to innovators. This will eventually lead to further technological and commercial success of OSS projects and sustainability of its development and growth over time.</p>
Breakdown of the concept of <i>flexibility</i>	<p data-bbox="555 1140 1382 1498">It is a challenging task for OSS firm's managers and especially the sales team to enlighten enterprise clients having little experience about issues like the <i>genuine benefits</i> of adopting OSS solutions. The mainstream perception of OSS solution is that it is a less costly solution when compared to proprietary ones (i.e., a pecuniary perspective). However, by unpacking the flexibility concept and revealing its various conceptual elements, firms can successfully inform clients of the real benefits such as influencing technology trajectories, and strategizing their business. They can explain what the intrinsic value of flexibility is and how it provides real customer benefits.</p>
Meaningful cost saving	<p data-bbox="555 1549 1382 1806">Cost saving is often perceived as a key benefit sought by clients adopting OSS solutions. However, as this research has demonstrated, there is more to OSS than this virtue. The open architecture of OSS which allows for maintainability and sustainability provide a meaningful cost saving benefit. It is one that emerges by remaining connected and interacting with the community of developers. Meaningful cost savings are thus not limited to acquisition costs, but also operation and maintenance (e.g., modification, switching) costs.</p>

---

Client typology and the nature of the client technological collaborations

This research's findings show that the degree and mode of collaboration and connection seem to influence the proposed relationship between adoption of OSS and the accrued key benefits such as flexibility and cost saving. Based on the nuances that differentiate clients from one another, the 'client typology' (Figure 5.11) informs OSS firms' managers and leadership of communities to be aware of differences which exist among enterprise clients and their expectations. Therefore, the levels of commitment and investments, as well as modes of collaboration are all influenced by the type of enterprise client, OSS firms and communities deal with. Therefore, they can benefit from these insights, and adapt their interactions to clients' particularities. In case of long-term planning, dual-purpose leaders can use these insights to train and shift a position of a client from, for example, 'the conservatives' to 'the champions' in order to enhance their involvements.

Context for, and modes of collaboration

These two issues are particularly insightful to those OSS firms and clients that are planning to enter into a long-term OSS technological collaboration over several projects. By knowing about the *key contextual elements*, and their *consequences* as well as *types of collaborations* and *relationships*, both OSS firms and clients can make informed decisions about how to strategize their involvement and resources allocations from the start.

Success and sustainability

Knowing better about these fuzzy notions is significant to managers of OSS firms, OSS community leadership, and managers of client organizations for at the end of the day, not all factors associated with investments can be calculated through a cost-benefit analysis. This is because 'value' of a specific project, module, or tool differs from firm to firm and individuals from business world or OSS technical world. There are different understandings of success and sustainability of OSS projects, its R&D process, and collaborations. Therefore, by better knowing the key elements associated with success and sustainability, clients, OSS firms, and community core developers can have meaningful and clear discussions with one another particularly when they want to forge *triadic relationships*.

#### Dual-purpose leadership

If actors, collaboration and relationships form the *bricks* of collaborative OSS R&D and innovation process (RDIP), the leadership is the *mortar* that glues all these constituents of OSS ecosystem together. The significance of leadership is further underlined in collaborations because within OSS projects the people in power positions are required to manage *technology* (and all associated issues) and *people* simultaneously in order to create synergy and realize the platform's key objectives. The emerging concept of 'dual-purpose leadership' therefore is comprised of five core categories (vision, goal setting, technology management, managing the fear of forking, and managing/leading the OSS community of actors) as well as a number of micro conceptual building blocks (e.g., coordination mechanisms, open governance, etc.). Delineating these concepts (Chapter 5, Section 5.3.2, Part D) therefore raises the level of awareness of managers about the *importance of establishing a collective leadership and playing an influential role* in the leadership process. Figure 5.15 demonstrates the institutional evolution of OSS and shows that OSS RDIP is evolving towards Open Technology Platforms. Therefore, organizations, communities and individuals all together are now viewed as stakeholders that can invest in their leadership efforts to make sure their technological self-interest is aligned with that of the group interest.

---

Source: The author.

## 7.5 Limitations of the study

This research has set out to *explore*, *describe* and *explain* the phenomenon of OSS technological collaborations and the shared R&D and innovation process that hosts them. Despite the insights generated within the scope of this research, the final work still suffers from several shortcomings. Particularly, the study's exploratory nature has made the research vulnerable to some criticism. For instance, according to Babbie (1999, p. 73), an *exploratory* research's findings may "seldom provide satisfactory answers to research questions, though they can hint at answers and give insights into the research methods that could provide definitive answers" (Babbie, 1999, p. 73). Thus, as this research is not an exception, in what follows, I discuss its major limitations and underline some avenues for further research.

### 7.5.1 Research logistics and socio-cultural related issues

Part of the research logistics includes the challenging task of preparing the list of potential participating candidates, obtaining their consent, and coordinating the schedules of these participants for interview sessions. In addition, having completed the interviews, I had to reach out to individuals to further reduce ambiguity, and avoid misrepresentations, as well as to obtain their approval of the transcriptions of their interview sessions. The number of interviews was limited due to a number of practical constraints.

First, because the participants were mostly individuals holding managerial positions (e.g., CEOs, VPs, project managers, entrepreneurs-founders, and R&D managers, to name a few), their time availability was limited (an interview session lasts for about 45 to 60 minutes), and they had to schedule, reschedule, cancel, and reinstate the interview sessions as they found it convenient to their daily routine. In some cases,

the interviews were conducted with managers stationed outside Canada, therefore time difference played yet another role in interview coordination task.

Second, although each interviewee received a consent form to sign, some managers (mostly those from large corporations) were not comfortable with the idea of having a conversation with an outsider – there was a *concern for confidentiality*. In one case, a manager signed the consent form, conducted the interview, even sent a copy of the consent form to the company's headquarter to be kept as a record. However, unfortunately, the legal department advised him to cancel everything and request the researcher to refrain from using the data from the interview session out of the fear of competition. Some interviewees prefer to err on the side of caution, meaning that they could be providing selected details of the true reality. Such behavior is entirely understandable. In fact, since the adoption of OSS solutions can tweak clients' business models and create strategic advantages, and because questions of the explanatory nature tend to ask for underlying details which can reveal some clients' secrets, the interviewees tend to shy away from the whole story. This issue coupled with the fact that interview session can sometimes lead to heated discussions, and the interviewee – who is principally an extremely technical person in managerial position – may lose control of what she or he can and cannot say makes it difficult for large corporation to be open to receiving strangers in their top management offices. Thus, these kinds of issues have imposed some limitations on data collection process (specifically in case of large organizations).

Third, OSS people are people from a wide range of technical backgrounds and at times specific OSS socio-cultural issues bind them together as a people. In general, based on the discussions put forth in Chapter 3 about free software and OS software, one can recognize that OS world is more leaning towards sharing, solidarity, free access and consumption which can be cautiously grouped together as a more

socialistic world. In fact, this unique socialist stance seen in OSS communities, makes a clear distinction between a world of sharing gratis (Free and OS world), and a world of selling the permission to use through license fees and royalties (World of Capitalism). Consequently, the business management researchers, like me, with no prior track record of contributing to OS world through code sharing and using free software solutions such as LibreOffice, FFmpeg, and Linux-based OS, may be looked upon as people who at least support and promote the capitalist mentality by using proprietary software products and paying license fees and royalties to Microsoft, and Apple. Put simply, I am so much of an outsider to this group of elite and dedicated OSS developers. For instance, during my interviews, I came across the top-notch OSS developers whose formal education backgrounds were in music, philosophy, sociology, and even law to name a few. Even so, out of extreme passion for OSS, they mastered the trade through a self-learning journey over a decade-long practice. Some interviewees were very much concerned with ethos of free software movement, some were more moderate, and some others were very much liberal and even open to coexistence between proprietary and free software.

All these factors made me realize that I was dealing with a sensitive and heterogeneous audience. Thus, I had to tailor my strategies to connect, and conduct the interviews differently. For instance, since the first day I started my participant observation, I started using OSS solutions such as LibreOffice which is an OS version of Microsoft word, and tried to get a sense of what it means to be part of OSS culture. Sometimes, to make meaningful contacts with some potential candidates, I had to find a trusted third person to open the door for me. Yet in some cases, I had to attend a seminar or a workshop on OSS to become familiar with the target person and build the necessary rapport before making any request for an interview. So, overall, although I received a tremendous amount of support during this research and met



with a lot of nice chaps, being an outsider to OS world was very challenging to get the interviews done.

All these three factors: participants' managerial position, confidentiality, and OSS socio-cultural values imposed a number of challenges on data collection process— not to mention the obvious issues such energy, time and expenses lost due to extra administrative works, rushing to interviews on short notices, and getting back to office disappointed due to last minute cancellations. Therefore, I had undergone enormous amount of administrative and logistics work to get each individual into the final stages of interview and post-interview follow-up.

However, despite all these issues, I managed to benefit greatly from the connections and friendships I developed through adoption of the participant observation method. With the help of the support of OSS activists within the Savoir-faire Linux Inc., I managed to obtain new and information-rich informants to proceed and complete my research.

#### 7.5.2 The researcher's theoretical sensitivity: Merits and demerits

*Theoretical sensitivity* refers to the personal qualities, background, and experience that a researcher applies to the research and underlying data. Glaser (1978) and Strauss and Corbin (1990) require a researcher to reveal his “theoretical sensitivity” for the analytical process of grounded theory to be effective. In fact, a researcher's background can be a source of bias and prejudice clouding his impartiality in interpreting the data.

In my case, lack of technical background in the field of software development and in particular OSS was both advantageous and disadvantageous. Being an individual from a social science background was a merit mainly because I did not have a frame

of reference both technical and conceptual to dictate me a preference for how things must be or how events shall happen. Such lack of specific reference helped me remain impartial throughout the research. To better grasp this issue, only imagine the situation where I was an ideological free software developer or an OSS activist studying OSS phenomenon! From day one in the field, I could only view OSS from a favorable and supportive lens, thereby leaving no room for adopting a critical perspective.

On the other hand, having a low level of theoretical sensitivity can be a limitation, specifically in high-tech arena such as software. Software world, and OSS world in particular, is replete with technical jargon. Respondents, “the [weird] technical guys” use very specific terms and examples to highlight their points. Sometimes you may get lost in the technical details and specifics to the extent that main topic is forgotten. After all, a technical person, a top-notch coder is amazed about the deep technological intricacies. It is therefore my job to make sense of details and classify them in an understandable and meaningful manner. And obviously, I am not a software engineer. Nevertheless, that does not mean I cannot *study* OSS phenomenon. It is just a more challenging task to me.

Fortunately, under the supervision of my industry advisor— Mr. Jérôme Oufella (VP Technologies at Savoir-faire Linux Inc.)— I managed to catch up with the technicalities and the nitty-gritty involved in OSS R&D and innovation process through my industry internship period. For one year, I spent about two to three days a week at the company interacting with software engineers from different departments and domains (e.g., product engineering, enterprise solutions, infrastructure, cloud architecture and big data analytics). Thankfully, I managed to add OSS-related savoir-faire to my toolbox and complete this research endeavor.

### 7.5.3 Challenges caused by abstraction of inductive theory building

Inductive theory building, and more particularly grounded theory approach work by building conceptual categories from higher levels of abstraction. Therefore, in the process of categorizing events and activities to reach that higher level of abstraction, the researcher may experience a potential loss of connection with reality (see Moffett, 1968). More precisely, Paré and Smart (1994, p. 153) emphasize that “whenever we create categories for analysis ‘we trade a loss of reality for a gain in control’”. Since the objective of this research is to develop an inductive theory, there is the potential for the researcher to introduce a bias unintentionally through grouping conceptual categories based on similarities and losing the opportunity to notice differences.

To overcome this challenge, I have tried to increase my theoretical sensitivity by doing the open coding process in two stages. First, I conducted open coding (i.e., mainly ‘line-by-line analysis’) while working in the field collecting the data. Next, once I collected all data, I re-engaged with data analysis and conducted a month-long open coding session parallel to axial coding. Since I gained more familiarity with the data and had already investigated the confusing points and clarified the dubious issues and textual errors, the second coding session became more meaningful. I gained a bird’s eye view (a gestalt perspective) of the data, and consequently, employing an iterative constant comparative method helped me to come across categories and relationships that were clearly distinct from one another in nature. However, I must acknowledge that, still, if two different researchers use the same data they may come up with different categories. This is part of the inherent subjectivity associated with grounded theory approach.

## 7.6 Avenues for further research

First, follow-up research is required to empirically test the proposed relationships – hypotheses– theorized based on the theoretical model (Figure 5.17) discussed in Chapter 5 (Section 5.5). Although my qualitative analysis of the data and the follow-up theoretical discussions in form of four core propositions (see Table 5.7) suggest the role (i.e., intervening and moderating) of ‘dual-purpose leadership’ and the moderating role of ‘contextual factors’, complementary large scale quantitative studies (i.e., survey study) can further prove or disprove these claims. We need to know about the statistical significance of the theoretical concepts introduced here, such as the dual-purpose leadership. Furthermore, in this study, I suggest a covariance effect between the two constructs of success and sustainability. This, however, seems logical on theoretical level based on the emerging concepts, only a theory testing approach can provide some confirmation about such claim.

Second, our case study approach to inductive theory building only allows for making *theoretical* or *analytical* generalizations (Yin, 2011, 2009). In fact, case study is concerned with analytic generalization where the researcher “is striving to generalize a particular set of results to some broader theory” (Yin, 2009, p. 43). Therefore, to establish *statistical* generalization and make inferences about a population based on sampled data, further survey research –which is a common method for making such generalization (Yin, 2011, 2009; Fowler, 1988) – is required.

Third, past research has not been sensitive to distinguishing between the nature and specific characteristics of firms in connecting with and benefiting from OSS projects and communities. This research clearly distinguishes among OSS firms, enterprise clients, hobbyists, and core development teams and their roles in the collaborative innovation process. I suggest that the diversity of actors (including enterprise clients) – under certain conditions (i.e., context for collaborations: know-who, know-how,

know-why)– is likely to positively influence both the sustainability and success of collaborative OSS RDIP (see proposition one in Table 5.7). Moreover, distinctions are made among OSS firms and enterprise clients concerning their strategies, levels of their technological and resources capabilities, and their influence on the level of their involvement. However, we need larger number of cases to analyze in order to better describe, and theorize these categories, and establish their boundaries in detail. Thus, follow-up qualitative research can build upon the perspectives offered in this work and enrich them through providing complementary or even contradictory evidences gathered through larger number of case studies.

Fourth, Chapter 6 (Table 6.2), to the best of my knowledge, is the first conceptual attempt to develop an alternative theoretical perspective of OSS technological collaborations vis-à-vis conventional strategic alliances and informal cooperation. Further theoretical and empirical evidence needs to be collected to complement the qualitative and conceptual data gathered in this research and consequently strengthen the proposed conceptual claims.

## APPENDIX

### A SAMPLE OF THE INTERVIEW QUESTIONS

1. How did you become familiar and eventually involved with OSS and OSS communities?
2. Why do you interact with OSS communities?
3. How do you collaborate with OSS communities?
4. How do you develop solutions collaboratively?
5. How do you make decision with regards to what code to share and what code to hold back in the R&D process?
6. In your opinion, how can one motivate OSS developers to keep working on an OSS project (long-term)?
7. How does OSS methodology impact the quality of OS software being developed?
8. In your opinion, do you think sharing code back with community and making investments in building a relationship with its members worthy or can be justified economically? Please elaborate on your reasons.
9. Please talk about clients of OSS firms. Who are they and how they are involved in the OSS development process?
10. How do you view the role of enterprise clients in the OSS R&D process?
11. How do you view the nature of leadership, and its role in the collaborative software development process?
12. How do you see the interactions between OSS firms and proprietary software developers?
13. In your opinion, what are the main disadvantages of community-based OSS solutions and OSS in general?

14. In your opinion, what kind of capabilities should an engineer have to work in a place like an OSS firm? (if we believe the proprietary software firm is different from OSS firm)
15. How do you view sustainability issue within OSS ecosystem? I mean, when do you think an OSS project is a sustainable one? What are the contributing factors and what are the challenges, based on your personal experience?
16. Does your OS company have enough talented people to do the job or do they really need the community?
17. How do you describe the relationship between your company and OS communities?
18. Do you think, or is there a possibility that in the long-run customers influence the technology trajectory of OSS communities?
19. How do you build trust to convince your clients to migrate to a risky solution such as OSS or at least as it appears to be risky?
20. Who are the contributors to your OS community? How do they interact with each other in research and project development phases?
21. How is the innovation process organized?
22. If somebody wants to grow in your community, is there a chance for new leaders to emerge in your context? What is the process of becoming a maintainer in your community?
23. How do you perceive project forking? What are the positive and negative aspects of it?
24. How does the following scenario impact your community? For example, if a private enterprise forks your project and builds on top of its core codes and commercializes it in its own product line. Does that diminish your community?
25. What is a healthy community?
26. What do you mean by “we sell expertise”? What is “expertise”?

27. How do they teach operating systems in Canadian educational system?
28. Do you believe the OSS developer and proprietary one have different set of skills?
29. How do you evaluate a comment made by some OSS firms regarding the issue that “unless there is a commercial end user involved, testing will not be taken seriously”?
30. Do you think some very professional OS communities eventually in the long-term end up organizing innovation process like proprietary software firms?
31. Do community leaders have business instincts?
32. How are innovations organized within community of developers?
33. How do you manage diversity within your community of developers?
34. How did you come to the point you decided to commercialize this OSS project?
35. What types of licenses you use for this project?
36. Have you had the case when client come directly to you as community instead of going to mainstream consultancy firms? Could you please talk about some of these instances?
37. How do you keep your community members motivated? How do you keep them doing what they are doing? And to also become more innovative?
38. Do you have proprietary software competing with your OSS?
39. What does your business model look like and how does OSS adoption fit into that scenario?
40. Do you engage with OS communities based on a long-term vision?
41. Do you have this idea on mind that you could or should play a leadership role while interacting and collaborating with OS communities?
42. Why did you decide to adopt OSS solutions? And why did you choose this specific partner for the migration process?
43. What is value or value-added OSS to your business line?



## REFERENCES<sup>92</sup>

- Abrahamson, E. (1996). Management fashion. *Academy of Management Review*, 21, 254–285.
- Adler, P. S., & Kwon, S. (2002). Social capital: Prospects for a new concept. *Academy of Management Review*, 21(1), 17–40.
- Afuah, A. (2004). *Business models: A strategic management approach*. New York: Irwin/McGraw-Hill.
- Afuah, A., & Tucci, C. L. (2001). *Internet business models and strategies: Text and cases*. New York: McGraw-Hill.
- Agranoff, R., & McGuire, M. (2003). *Collaborative public management: New strategies for local governments*. Washington, DC: Georgetown University Press.
- Ahuja, G. (2000). Collaboration networks, structural holes, and innovation: A longitudinal study. *Administrative Science Quarterly*, 45(3), 425–455.
- Aiken, M., & Hage, J. (1972). Organizational permeability. Paper presented at the 1972 Meetings of the American Sociological Association.
- Aiken, M., & Hage, J. (1968). Organizational interdependence and intraorganizational structure. *American Sociological Review*, 33, 912–930.
- Ajila, S. A., & Wu, D. (2007). Empirical study of the effects of open source adoption on software development economics. *The Journal of Systems and Software*, 80, 1517–1529.
- Aksoy-Yurdagul, D. (2015). The impact of open source software commercialization on firm value. *Industry and Innovation*, 22(1), 1–17.

---

<sup>92</sup> Bibliography is formatted according to the sixth edition of the *Publication Manual of the American Psychological Association* (BF 76.7 .P83 2010).

- Aldrich, H., & Herker, D. (1977). Boundary spanning roles and organization structure. *The Academy of Management Review*, 2(2), 217-230.
- Aldrich, H., & Pfeffer, J. (1976). Environment of organizations. In A. Inkeles (Ed.), *Annual Review of Sociology* (pp. 79-105), Vol. II, Palo Alto, Calif.: Annual Review Inc.
- Aldrich, H. E. (1979). *Organizations and environments*. Englewood Cliffs, NJ: Prentice-Hall.
- Alexy, O., & George, G. (2013). Category divergence, straddling, and currency: Open innovation and the legitimation of illegitimate categories. *Journal of Management Studies*, 50(2), 173-203.
- American Psychological Association. (2010). *Publication manual of the American Psychological Association* (6<sup>th</sup> ed.). Washington, DC: Author.
- Amit, R., & Schoemaker, P. (1993). Strategic assets and organizational rent. *Strategic Management Journal*, 14, 33-46.
- Amit, R., & Zott, C. (2001). Value creation in e-business. *Strategic Management Journal*, 22(2), 493- 520.
- Amrit, C., & van Hillegersberg, J. (2010). Exploring the impact of socio-technical core-periphery structures in open source software development. *Journal of Information Technology*, 25, 216–229.
- Applegate, L. M. (2001). Emerging e-business models. *Harvard Business Review*, 79(1), 79–87.
- Appleyard, M., & Chesbrough, H. W. (2016). The dynamics of open strategy: From adoption to reversion. *Long Range Planning*, xx(xx), xx-xx.
- Argote, L., McEvily, B., & Reagans, R. (2003). Managing knowledge in organizations: An integrative framework and review of emerging themes. *Management Science*, 49, 571–82.

- Andrews, K. R. (1980). *The concept of corporate strategy*. Irwin: Homewood.
- Arora, A., Fosfuri, A., & Gambardella, A. (2001). *Markets for technology: The economics of innovation and corporate strategy*. Cambridge: MIT Press.
- Arrow, K. (1962). The economic implications of learning by doing. *Review of Economic Studies*, 29(3), 155-173.
- Arthur, W. B. (1994). *Increasing returns and path dependence in the economy*. US: University of Michigan Press, Ann Arbor.
- Arthur, W. B. (1989). Competing technologies, increasing returns, and lock-in by historical events. *Economic Journal*, 99, 116-131.
- Arthur, W. B. (1996). Increasing returns and the new world of business. *Harvard Business Review*, 7(4), 100-109.
- Arthur, W. B. (1997). Path-dependence, self-reinforcement, and human learning. In W.B. Arthur (Ed.), *Increasing returns and path dependence in the economy* (pp. 133-158). Ann Arbor, MI: The University of Michigan Press.
- Asay, M. (2014, May 6). Open source failure is its greatest success. *TechRepublic*. Retrieved from <http://www.techrepublic.com/article/open-source-failure-is-its-greatest-success/>
- Aspara, J., Lamberg, J-A., Laukia, A., & Tikkanen, H. (2013). Corporate business model transformation and inter-organizational cognition: The case of Nokia. *Long Range Planning*, 46, 459-474.
- Babbie, E. R. (1999). *The practice of social research* (Revised edition of 8<sup>th</sup> ed.). Belmont, CA: Wadsworth.
- Bacharach, S. B. (1989). Organizational theories: Some criteria for evaluation. *Academy of Management Review*, 14(4), 496-515.
- Bacharach, S. B., & Lawler, E. J. (1981). Power and tactics in bargaining. *ILR Review*, 34(2), 219-233.

- Badawy, M. K. (2011). "Is open innovation a field of study or a communication barrier to theory development?": A perspective. *Technovation*, 31(1), 65–67.
- Baden-Fuller, C., & Haefliger, S. (2013). Business models and technological innovation. *Long Range Planning*, 46, 419-426.
- Baden-Fuller, C., Morgan, M. (2010). Business models. *Long Range Planning*, 43, 156–171.
- Bagozzi, R. P., & Dholakia, U.M. (2006). Open source software user communities: A study of participation in Linux user groups. *Management Science*, 52(7), 1099–1115.
- Baker, G., Gibbons, R., & Murphy, K. (2002). Relational contracts and the theory of the firm. *Quarterly Journal of Economics*, 117, 39–83.
- Balconi, M., Brusoni, S., & Orsenigo, L. (2010). In defence of the linear model: An essay. *Research Policy*, 39, 1-13.
- Banik, M., & Taherizadeh, A. (2014, May 27-30). Investigating the Borromean Trinity of open source software innovation: Case study research for theory building on open innovation. *Industry Studies Conference – Sustaining Sustainability in the Face of Increasing Risk*, Portland, U.S.A.
- Barge-Gil, A. (2013). Open strategies and innovation performance. *Industry and Innovation*, 20(7), 585-610.
- Barney, J. B. (1986). Strategic factor markets: expectations, luck, and business strategy. *Management Science*, 32(10), 1231-1241.
- Barney, J. B. (1999). How firm capabilities affect boundary decisions. *Sloan Management Review*, 40(1), 19-32.
- Barney, J. (1991). Firm resources and sustained competitive advantage. *Journal of Management*, 17, 99–120.

- Becker, H. S. (1958). Problems of inference and proof in participant observation. *American Sociological Review*, 23, 652-660.
- Becker, H. S., & Geer, B. (1957). Participant observation and interviewing: A comparison. *Human Organization*, 16, 28-32.
- Benkler, Y. (2002). Coase's penguin, or, Linux and "the nature of the firm". *The Yale Law Journal*, 112(3), 369-446.
- Benkler, Y. (2006). *The wealth of networks: How social production transforms markets and freedom*. New Haven, Conn: Yale University Press.
- Bergquist, M., & Ljungberg, J. (2001). The power of gifts: Organizing social relationships in open source communities. *Information Systems Journal*, 11, 305-320.
- Berkholz, D. (2014). *Donnie Berkholz's story of data- Making sense out of information*. Retrieved from RedMonk Information website: <http://redmonk.com/dberkholz/2014/05/02/github-language-trends-and-the-fragmenting-landscape/#ixzz30wEgsUif>
- Bikhchandani, S., Hirshleifer, D., & Welch, I. (1992). A theory of fads, fashion, custom, and cultural change as informational cascades. *Journal of Political Economy*, 100(5), 992-1018.
- Bitzer, J., Schrettl, W., & Schröder, P. J. H. (2007). Intrinsic motivation in open source software development. *Journal of Comparative Economics*, 35, 160-169.
- Bizzi, L., & Langley, A. (2012). Studying processes in and around networks. *Industrial Marketing Management*, 41, 224-234.
- Björkdahl, J. (2009). Technology cross fertilization and the business model: The case of integrating ICTs in mechanical engineering products. *Research Policy*, 38, 1468-1477.

- Blazevic, V., & Lievens, A. (2008). Managing innovation through customer coproduced knowledge in electronic services: An exploratory study. *Journal of the Academy of Marketing Science*, 36(1), 138-151.
- Blumer, H. (1969). *Symbolic interaction*. Englewood Cliffs, NJ: Prentice-Hall.
- Bogers, M., & Lhuillery, S. (2011). A functional perspective on learning and innovation: Investigating the organization of absorptive capacity. *Industry and Innovation*, 18(6), 581-610.
- Bogers, M., & West, J. (2012). Managing distributed innovation: Strategic utilization of open and user innovation. *Creativity and Innovation Management*, 21(1), 61-75.
- Bonaccorsi, A., & Rossi, C. (2003). Why Open Source software can succeed. *Research Policy*, 32, 1243-1258.
- Bonaccorsi, A., & Rossi, C. (2006). Comparing motivations of individual programmers and firms to take part in the open source movement: From community to business. *Knowledge, Technology and Policy*, 18(4), 40-64.
- Bonaccorsi, A., Giannangeli, S., & Rossi, C. (2006). Entry strategies under competing standards: Hybrid business models in the open source software industry. *Management Science*, 52(7), 1085-1098.
- Borgatti, S. P., & Everett, M. G. (1999). Models of core/periphery structures. *Social Networks*, 21, 375-395.
- Boudreau, K. J., & Hagiu, A. (2009). Platform rules: Multi-sided platforms as regulators. In Annabelle Gawer (Ed.), *Platforms, markets and innovation* (pp. 163-191). Cheltenham, UK: Edward Elgar Publishing.
- Bourdieu, P. (1983). Forms of capital. In J.G. Richardson (Ed.), *Handbook for theory and research for the sociology of education* (pp. 241-258). New York: Greenwood.
- Brandenburger, A. M., & Nalebuff, B. J. (1996). *Co-opetition*. Boston, MA: Harvard Business School Press.

- Brandenburger, A. M., & Stuart, H. W., Jr. (1996). Value-based business strategy. *Journal of Economics and Management Strategy*, 5(1), 5-24.
- Brannen, J. (1992). Combining qualitative and quantitative approaches: An overview. In J. Brannen (Ed.), *Mixing methods: Qualitative and quantitative research* (pp. 3–37). Aldershot: Avebury.
- Brusoni, S., Prencipe, A., & Pavitt, K. (2001). Knowledge specialisation, organizational coupling and the boundaries of the firm: Why firms know more than they make? *Administrative Science Quarterly*, 46(4), 597–621.
- Bryman, A. (2004). *Social research methods* (2<sup>nd</sup> ed.). Oxford: Oxford University Press.
- Bryman, A. (2006). Integrating quantitative and qualitative research: How is it done? *Qualitative Research*, 6(1), 97-113.
- Buchanan, J. M. (2001). Game theory, mathematics, and economics. *Journal of Economic Methodology*, 8, 27-32.
- Bucherer, E., Eisert, U., & Gassmann, O. (2012). Towards systematic business model innovation: Lessons from product innovation management. *Creativity and Innovation Management*, 21(2), 183-198.
- Burrell, G., & Morgan, G. (1979). *Sociological paradigms and organizational analysis*. London: Heinemann.
- Calia, R. C., Guerrini, F. M., & Moura, G. L. (2007). Innovation networks: From technological development to business model reconfiguration. *Technovation*, 27, 426-432.
- Campbell, D. (1969). Variation and selective retention in socio-cultural evolution. *Gen. Sys.* 16, 69-85.
- Casadesus-Masanell, R., & Ghemawat, P. (2006). Dynamic mixed duopoly: A model motivated by Linux vs. Windows. *Management Science*, 52(7), 1072–1084.

- Casadesus-Masanell, R., & Ricart, J. E. (2010). From strategy to business models and onto tactics. *Long Range Planning*, 43(2), 195–215.
- Casciaro, T., & Piskorski, M. J. (2005). Power imbalance, mutual dependence, and constraint absorption: A closer look at resource dependence theory. *Administrative Science Quarterly*, 50, 167-199.
- Cassiman, B., & Veugelers, R. (2002). R&D cooperation and spillovers: Some empirical evidence from Belgium. *American Economic Review*, 92(4), 1169-1184.
- Cassiman, B., & Veugelers, R. (2006). In search of complementarity in the innovation strategy: internal R&D and external knowledge acquisition. *Management Science*, 52(1), 68–82.
- Chalmers, A. (1982). *What is this thing called science?* Queensland, Australia: University of Queensland.
- Chan, S., Kensinger, J., Keown, A., & Martin, J. (1997). Do strategic alliances create value? *Journal of Financial Economics*, 46, 199–221.
- Chanal, V., & Caron-Fasan, M.-L. (2010). The difficulties involved in developing business models open to innovation communities: The case of a crowdsourcing platform. *M@n@gement*, 13(4), 318–341.
- Chandler, A. D. (1962). *Strategy and Structure: Chapters in the history of industrial enterprise*. USA: MIT Press.
- Chandler, A. D. (1990). The enduring logic of industrial success. *Harvard Business Review*, March/April, 130-140.
- Charmaz, K. (2006). *Constructing grounded theory: A practical guide through qualitative analysis*. Thousand Oaks, CA: SAGE.



- Charmaz, K. (2011). Grounded theory methods in social justice research. In N. K. Denzin & Y.S. Lincoln (eds.), *Handbook of qualitative research* (14<sup>th</sup> ed., pp. 359-380). Thousand Oaks, CA: SAGE.
- Chesbrough, H., (2003a). *Open innovation: the new imperative for creating and profiting from technology*. Boston: Harvard Business School Press.
- Chesbrough, H., (2003b). The logic of open innovation: Managing intellectual property. *California Management Review*, 45(3), 33-58.
- Chesbrough, H., (2003c). The era of open innovation. *MIT Sloan Management Review*, 44(3), 35-41.
- Chesbrough, H. W. (2006a). *Open business models: How to thrive in the new innovation landscape*. Boston, MA: Harvard Business School Press.
- Chesbrough, H. W. (2006b). Open innovation: A new paradigm for understanding industrial innovation. In H. W. Chesbrough, W. Vanhaverbeke, & J. West (Eds.), *Open innovation: researching a new paradigm* (Vol. 21, pp. 1-12). New York: Oxford University Press.
- Chesbrough, H. W. (2006c). *Open innovation: The new imperative for creating and profiting from technology*. Paperback edition. Harvard Business School Press.
- Chesbrough, H. (2007a). Business model innovation: It's not just about technology anymore. *Strategy & Leadership*, 35(6), 12-17.
- Chesbrough, H. (2007b). Why companies should have open business models. *MIT Sloan Management Review*, 48(2), 22-28.
- Chesbrough, H. (2010). Business model innovation: Opportunities and barriers. *Long Range Planning*, 43, 354-363.
- Chesbrough, H. (2011). *Open service innovation: Rethinking your business to grow and compete in a new era*. San Francisco, CA: Jossey Bass.

- Chesbrough, H., & Appleyard, M. (2007). Open innovation and strategy. *California Management Review*, 50(1), 57-76.
- Chesbrough, H., & Bogers, M. (2014). Explicating open innovation clarifying an emerging paradigm for understanding innovation. In Henry W. Chesbrough, Wim Vanhaverbeke, Joel West (Eds.), *New frontiers in open innovation* (pp.1- 28). Oxford: Oxford University Press.
- Chesbrough, H., & Crowther, A. K. (2006). Beyond high tech: Early adopters of open innovation in other industries. *R&D Management*, 36(3), 229-236.
- Chesbrough, H., & Rosenbloom, R. S. (2002). The role of the business model in capturing value form innovation: Evidence from Xerox corporation's technology spin-off companies. *Industrial and Corporate Change*, 11(3), 529-555.
- Chesbrough, H. W., & Schwartz, K. (2007). Innovating business models with co-development partnerships. *Research-Technology Management*, 50(1), 55-59.
- Chesbrough, H., Vanhaverbeke, W., & West, J. (Eds.) (2006). *Open innovation: Researching a new paradigm*. Oxford, UK: Oxford University Press.
- Chi, T. (1994). Trading in strategic resources: Necessary conditions, transaction cost problems, and choice of exchange structure. *Strategic Management Journal*, 15, 271-290.
- Chou, Shih-Wei, & He, Mong-Young. (2011). The factors that affect the performance of open source software development –the perspective of social capital and expertise integration. *Information Systems Journal*, 21, 195-219.
- Christensen, C. (1997). *The innovator's dilemma*. Boston, MA: Harvard Business School Press.
- Christensen, C., & Rosenbloom, R. S. (1995). Explaining the attacker's advantage: Technological paradigms, organizational dynamics, and the value network. *Research Policy*, 24, 233-257.

- Christensen, J. F. (2014). Open innovation and industrial dynamics: Towards a framework of business convergence. In Henry Chesbrough; Wim Vanhaverbeke; Joel West (Eds.), *New frontiers in open innovation* (pp. 94-114). Oxford: Oxford University Press.
- Christensen, J-F., Olesen, M. H., & Kjær, J. S. (2005). The industrial dynamics of open innovation – evidence from the transformation of consumer electronics. *Research Policy*, 34, 1533–1549.
- Chu, P.-Y., & Chen, W.-C. (2011). Open business models: A case study of System-on-a-Chip (SoC) design foundry in the integrated circuit (IC) industry. *African Journal of Business Management*, 5(21), 8536–8544.
- Coase, R. H. (1937). The nature of the firm. *Economica*, 4(16), 386–405.
- Cohen, W. M., & Levinthal, D. A. (1990). Absorptive capacity: A new perspective on learning and innovation. *Administrative Science Quarterly*, 35, 128-152.
- Conner, K., & Prahalad, C. K. (1996). A resource-based theory of the firm: Knowledge versus opportunism. *Organization Science*, 7(5), 477-501.
- Constant, D., Sproull, L., & Kiesler, S. (1996). The kindness of strangers: The usefulness of electronic weak ties for technical advice. *Organization Science*, 7(2), 119–135.
- Creswell, J. W. (2003) *Research design: Qualitative, quantitative, and mixed methods approaches* (2<sup>nd</sup> edition). Thousand Oaks, CA: Sage.
- Crowston, K., & Scozzzi, B. (2002). Open source software projects as virtual organizations: Competency rallying for software development, *IEEE Proceedings Software*, 149(1), 3–17.
- Crowston, K., Annabi, H., & Howison, J. (2003). Defining open source software project success. *ICIS 2003 Proceedings*, 327-340.

- Crowston, K., Howison, J., & Annabi, H. (2006). Information systems success in free and open source software development: Theory and measures, *Software Process: Improvement and Practice*, 11(2), 123–148.
- Dahlander, L., Frederiksen, L. (2012). The core and cosmopolitans: A relational view of innovation in user communities. *Organization Science*, 23(4), 988-1007.
- Dahlander, L., Frederiksen, L., & Rullani, F. (2008). Online communities and open innovation. *Industry and Innovation*, 15(2), 115-123.
- Dahlander, L., & Gann, D. M. (2010). How Open Is Innovation? *Research Policy*, 39, 699-709.
- Dahlander, L., & Magnusson, M. G. (2005). Relationships between open source software companies and communities: Observations from Nordic firms. *Research Policy*, 34, 481-493.
- Dahlander, L., & Magnusson, M. G. (2008). How do firms make use of open source communities? *Long Range Planning*, 41(6), 629-649.
- Dahlander, L., & Wallin, M. W. (2006). A man on the inside: Unlocking communities as complementary assets. *Research Policy*, 35, 1243–1259.
- Das, S., Sen, P. K., & Sengupta, S. (1998). Impact of strategic alliances on firm valuation. *Academy of Management Journal*, 41, 27-41.
- Das, T. K., & Teng, B.-S. (2000). A resource-based view theory of strategic alliances. *Journal of Management*, 26(1), 31-61.
- Davey, S. M., Brennan, M., Meenan, B. J., & McAdam, R. (2010). The health of innovation: Why open business models can benefit the healthcare sector. *Irish Journal of Management*, 30(1), 21–40.
- Davey, S. M., Brennan, M., Meenan, B. J., & McAdam, R. (2011). Innovation in the medical device sector: An open business model approach for high-tech small firms. *Technology Analysis & Strategic Management*, 23(8), 807–824.

- David, P. A. (1985). Clio and the economics of QWERTY. *The American Economic Review*, 75(2), 332-337.
- David, P.A., & Rullani, F. (2008). Dynamics of innovation in an “open source” collaboration environment: Lurking, laboring, and launching FLOSS projects on SourceForge. *Industrial and Corporate Change*, 17(4), pp. 647–710.
- Davis, F. D. (1989). Perceived usefulness, perceived ease of use and user acceptance of information technology. *MIS Quarterly*, 13, 319–340.
- Dam, K. W. (1995). Some economic considerations in the intellectual property protection of software. *The Journal of Legal Studies*, 24(2), 321-377.
- Deglise, F. (2013, March 29). Québec adopte deux décrets contre le logiciel libre. *Le Devoir*. Retrieved from <http://www.ledevoir.com/politique/quebec/374522/quebec-adopte-deux-decrets-contre-le-logiciel-libre>.
- Delamont, S. (2004). Ethnography and participant observation. In C. Delamont, G. Giampietro, J. F. Gubrium, & D. Silverman (Eds.), *Qualitative research practice* (pp. 217-229). London: SAGE.
- DeLone, W. H., & McLean, E. R. (1992). Information systems success: The quest for the dependent variable. *Information Systems Research*, 3(1), 60–95.
- DeLone, W. H., & McLean, E. R. (2002). Information systems success revisited. In *Proceedings of the 35th Hawaii International Conference on System Sciences (HICSS-35)*, Waikola, Hawaii. Retrieved from <http://csdl2.computed.org/comp/proceedings/hicss/2002/1435/08/14350238.pdf>
- DeLone, W. H., & McLean, E. R. (2003). The DeLone and McLean model of information systems success: A ten-year update. *Journal of Management Information Systems*, 19(4), 9–30.
- Demil, B., & Lecocq, X. (2010). Business model evolution: In search of dynamic consistency. *Long Range Planning*, 43, 227-246.

- Dempsey, B. J., Weiss, D., Jones, P., & Greenberg, J. (2002). Who is an open source software developer? *Communications of the ACM*, 45(2) 67–72.
- Demsetz, H. (1967). Towards a theory of property rights. *American Economic Review*, 57, 347-359.
- Denzin, N. K. (1978). *The research act* (2<sup>nd</sup> ed.). New York: McGraw-Hill.
- Denzin, N. K., & Lincoln, Y. S. (2000). Introduction: The discipline and practice of qualitative research. In N. K. Denzin, & Y. S. Lincoln (Eds.), *Handbook of qualitative research* (2<sup>nd</sup> ed., pp. 1-29). Thousand Oaks: SAGE.
- Denzin, N. K., & Lincoln, Y. S. (Eds.). (2011). *The SAGE handbook of qualitative research* (4<sup>th</sup> ed.). USA: SAGE Publications, Inc.
- Denzin, N. K., & Lincoln, Y. S. (2005). Introduction: The discipline and practice of qualitative research. In N. K. Denzin & Y. S. Lincoln (Eds.), *Handbook of qualitative research* (3<sup>rd</sup> ed., pp. 1-44). Thousand Oaks, CA: SAGE.
- DeWalt, K. M., & DeWalt, B. R. (2002). *Participant observation*. Walnut Creek, CA: Altamira Press.
- Diesing, P. (1971). *Patterns of discovery in the social sciences*. Chicago: Aldine.
- Dittrich, K., & Duysters, G. (2007). Networking as a means to strategy change: The case of open innovation in mobile telephony. *R&D Management*, 24(6), 510-521.
- Dodgson, M., Gann, D., & Salter, A. (2006). The role of technology in the shift towards open innovation: The case of Procter & Gamble. *R&D Management*, 36(3), 333- 346.
- Doganova, L., & Eyquem-Renault, M. (2009). What do business models do? Innovation devices in technology entrepreneurship. *Research Policy*, 38, 1559- 1570.

- Doz, Y. (1996). The evolution of cooperation in strategic alliances: Initial conditions or learning processes. *Strategic Management Journal*, Summer Special Issue, 17, 55–84.
- Drechsler, W., & Natter, M. (2012). Understanding a firm's openness decisions in innovation. *Journal of Business Research*, 65, 438–445.
- Duarte, V., & Sarkar, S. (2011). Separating the wheat from the chaff – a taxonomy of open innovation. *European Journal of Innovation Management*, 14(4), 435–459.
- Dyer, J. H. (1997). Effective interfirm collaboration: How firms minimize transaction costs and maximize transaction value. *Strategic Management Journal*, 18, 553–556.
- Dyer, J. H., & Nobeoka, K. (2000). Creating and managing a high performance knowledge-sharing network: The Toyota case. *Strategic Management Journal*, 21, 345–367.
- Dyer, J., & Singh, H. (1998). The relational view: Cooperative strategy and sources of interorganizational competitive advantage. *Academy Management Review*, 23(4), 660–79.
- Eisenhardt, K. M. (1989). Building theories from case study research. *Academy of Management*, 14, 532–550.
- Eisenhardt, K. M., & Martin, J. A. (2000). Dynamic capabilities: What are they? *Strategic Management Journal*, 21, 1105–1121.
- Eisenhardt, K. M., & Schoonhoven, C. B. (2000). Resource-based view of strategic alliance formation: Strategic and social effects in entrepreneurial firms. *Organization Science*, 7(2), 136 – 150.
- Elg, U. (2000). Firms' home–market relationships: Their role when selecting international alliance partners. *Journal of International Business Studies*, 31, 169–177.

- Elmquist, M., Fredberg, T., & Ollila, S. (2009). Exploring the field of open innovation. *European Journal of Innovation Management*, 12(3), 326 – 345.
- Enders, A., Hungenberg, H., Denker, H-P., Mauch, S. (2008). The long tail of social networking. Revenue models of social networking sites. *European Management Journal*, 26, 199–211.
- Enders, A., Konig, A., & Hungenberg, H. (2009). Towards an integrated perspective of strategy: The value-process framework. *Journal of Strategy and Management*, 2(1), 76-96.
- Enkel, E., Gassmann, O., & Chesbrough, H. (2009). Open R&D and open innovation: Exploring the phenomenon. *R&D Management*, 39(4), 311–316.
- Evered, R. (1983). So what *is* strategy? *Long Range Planning*, 16(3), 57-72.
- Faems, D., de Visser, M., Andries, P., & van Looy, B. (2010). Technology alliance portfolios and financial performance: Value-enhancing and cost-increasing effects of open innovation. *Journal of Product Innovation Management*, 27(6), 785–96.
- Fanelli, D. (2010). Do pressures to publish increase scientists' bias? An empirical support from US States data. *PLoS ONE* 5(4), 1-7. doi:10.1371/journal.pone.0010271.
- Fanelli, D. (2009). How many scientists fabricate and falsify research? A systematic review and meta-analysis of survey data. *PLoS ONE* 4(5), e5738. doi:10.1371/journal.pone.0005738.
- Felin, T., & Zenger, T. R. (2014). Closed or open innovation? Problem solving and the governance choice. *Research Policy*, 43(5), 914–925.
- Feller, J., Finnegan, P., Fitzgerald, B., & Hayes, J. (2008). From peer production to productization: A study of socially enabled business exchanges in open source service networks. *Information Systems Research*, 19(4), 475-493.



- Feller, J., & Fitzgerald, B. (2000). A framework for understanding the open source software development paradigm. *ICIS Conference Proceeding*, Atlanta Association for Information Systems, Atlanta, GA, 58–69.
- Fey C. F., & Birkinshaw, J. (2005). External sources of knowledge, governance mode, and R&D performance. *Journal of Management*, 31(4), 597–621.
- Fitzgerald, B. (2006). The transformation of open source software. *MIS Quarterly*, 30(3), 587–598.
- Folta, T. B. (1998). Governance and uncertainty: The trade-off between administrative control and commitment. *Strategic Management Journal*, 19, 1007–1028.
- Folta, T. B., & Miller, K. D. (2002). Real options in equity partnerships. *Strategic Management Journal*, 23, 77–88.
- Fosfuri, A., Giarratana, M. S., & Luzzi, A. (2008). The penguin has entered the building: The commercialization of open source software products. *Organization Science*, 19(2), 292–305.
- Fowler, F. J., Jr. (1988). *Survey research methods* (Rev. ed.). Newbury Park, CA: SAGE.
- Fox, M.F. (1992). Research, teaching, and publication productivity: Mutuality versus competition in academia. *Sociology of Education*, 65, 293–305.
- Franck, E., & Jungwirth, C. (2003). Reconciling rent-seekers and donators – The governance structure of open source. *Journal of Management and Governance*, 7, 401–421.
- Franke, N., & Shah, S. (2003). How communities support innovative activities: An exploration of assistance and sharing among end-users. *Research Policy*, 32, 157–178.

- Franke, N., & von Hippel, E. (2003). Satisfying heterogeneous user needs via innovation toolkits: The case of Apache security software. *Research Policy*, 32(7), 1199-1215.
- Freeman, C. (1974). *The economics of industrial innovation*. London: Pinter.
- Galaskiewicz, J. (1985). Interorganizational Relations. *American Review of Sociology*, 11, 281-304.
- Gallivan, M. J. (2001). Striking a balance between trust and control in virtual organizations: A content analysis of open source software case studies. *Information Systems Journal*, 11(4), 277-304.
- Garvey, G. (1995). Why reputation favors joint ventures over vertical and horizontal integration: A simple model. *Journal of Economic Behavior and Organization*, 28, 387-397.
- Gassmann, O., & Enkel, E. (2004). Towards a theory of open innovation: Three core process archetypes. *Proceedings of the R&D Management Conference*, Lisbon, Portugal, July 6-9.
- Gassmann, O. (2006). Opening up the innovation process: Towards an agenda. *R&D Management*, 36(3), 223-228.
- Gassmann, O., Enkel, E., & Chesbrough, H. (2010). The future of open innovation. *R&D Management*, 40(3), 213-221.
- Gawer, A., & Cusumano, M. A. (2014). Industry platforms and ecosystem innovation. *Journal of Product Innovation Management*, 31(3), 417-433.
- Gephart, R. P. (1997). Hazardous measures: An interpretive textual analysis of quantitative sense making during crises. *Journal of Organizational Behavior*, 18, 583-622.
- Gephart, R. P. (1999). *Paradigms and research methods*. Research Methods Division forum, 4. [aom.pace.edu/rmd/1999\\_RMD\\_Forum\\_Paradigms\\_and\\_Research\\_Methods.htm](http://aom.pace.edu/rmd/1999_RMD_Forum_Paradigms_and_Research_Methods.htm).

- Gerlach, M. L. (1992). The Japanese corporate network: A blockmodel approach. *Administrative Science Quarterly*, 37, 105-139.
- Ghosh, R. A. (1998). FM interview with Linus Torvalds: What motivates free software developers? *First Monday* 3(3). doi: <http://dx.doi.org/10.5210/fm.v0i0.1475>
- Glaser, B., & Strauss, A. (1967). *The discovery of grounded theory: Strategies for qualitative research*. London: Wiedenfeld and Nicholson.
- Glaser, B. (1978). *Theoretical sensitivity*. Mill Valley, CA: Sociology Press.
- Glass, R. (1999). Of open source, Linux and hype. *IEEE Software*, 16(1), 126–128.
- Goes, J. B., & Park, S. H. (1997). Interorganizational links and innovation: The case of hospital services. *Academy of Management Journal*, 40, 673-696.
- Gomes-Casseres, B. (1987). Joint venture instability: Is it a problem. *Columbia Journal of World Business*, 97-102.
- Gomes-Casseres, B., Hagedoorn, J., & Jaffe, A. B. (2006). Do alliances promote knowledge flows? *Journal of Financial Economics*, 80, 5-33.
- Gompers, P., & Lerner, J. (2001). The venture capital revolution. *The Journal of Economic Perspective*, 15(2), 145-168.
- Granovetter, M. (1985). Economic action and social structure: A theory of embeddedness. *American Journal of Sociology*, 91, 481–510.
- Granstrand, O. (1999). *The Economics and Management of Intellectual Property*. Cheltenham, UK: Edward Elgar.
- Grant, R. M. (1991). The Resource-based theory of competitive advantage: Implications for strategy formulation. *California Management Review*, 33(3), 114-135.

- Grant, R. M. (1996). Prospering in dynamically-competitive environments: Organizational capability as knowledge integration. *Organization Science*, 7(4), 375–87.
- Grant, R. M., & Baden-Fuller, C. (1995). A knowledge-based theory of inter-firm collaboration. *Best Paper Proceedings 1995, Academy of Management*, 17–21.
- Grant, R. M., & Baden-Fuller, C. (2004). A knowledge accessing theory of strategic alliances. *Journal of Management Studies*, 41(1), 61–84.
- Gray, P. H., & Cooper, W. H. (2010). Pursuing failure. *Organizational Research Methods*, 13(4), 620–643.
- Gray, B., & Wood, D. J. (1991). Collaborative alliances: Moving from practice to theory. *Journal of Applied Behavioral Science*, 27, 3–22.
- Grewal, R., Lilien, G. L., & Mallapragada, G. (2006). Location, location, location: How network embeddedness affects projects success in open source systems. *Management Science*, 52(7), 1043–1056.
- Grindley, P. (1995). *Standards, strategy and policy: Cases and stories*. New York: Oxford University Press.
- Groen, A. J., & Linton, J. D. (2010). Is open innovation a field of study or a communication barrier to theory development? *Technovation*, 30, 554.
- Gruber, M., & Henkel, J. (2006). New ventures based on open innovation – an empirical analysis of start-up firms in embedded Linux. *International Journal of Technology Management*, 33(4), 356–372.
- Guba, E. (1990). *The paradigm dialog*. Beverly Hills, CA: SAGE.
- Gulati, R. (1995). Does familiarity breed trust? The implications of repeated ties for contractual choice in alliances. *Academy of Management Journal*, 38, 85–112.

- Gulati, R. (1998). Alliances and networks. *Strategic Management Journal*, 19, 293-317.
- Haefliger, S., von Krogh, G., & Spaeth, S. (2008). Code reuse in open source software. *Management Science*, 54(1), 180-193.
- Hagedoorn, J. (2002). Inter-firm R&D partnerships: An overview of major trends and patterns since 1960. *Research Policy*, 31, 477-492.
- Hagedoorn, J. (1993). Understanding the rationale of strategic technology partnering: Interorganizational modes of cooperation and sectoral differences. *Strategic Management Journal*, 14(5), 371-85.
- Hallen, L., Johanson, J., & Seyed-Mohamed, N. (1991). Interfirm adaptations in business relationships. *Journal of Marketing*, 55(2), 29-37.
- Hambrick, D. C. (1983). High profit strategies in mature capital goods industries: A contingency approach. *Academy of Management Journal*, 26(4), 687-707.
- Hamel, G. (1990). *Competitive collaboration: learning, power and dependence in international strategic alliances*. Unpublished doctoral dissertation. Graduate School of Business Administration, University of Michigan.
- Hamel, G. (1991). Competition for competence and interpartner learning within international strategic alliances. *Strategic Management Journal*, 12, 83-103.
- Hang, J., Hohensohn, H., Mayr, K., & Wieland, T. (2005). Benefits and pitfalls of open source in commercial contexts. In S. Koch (Ed.), *Free/Open source software development* (pp. 222-241). Hershey, PA: Idea Group Publishing.
- Hannan, M. T., & Freeman, J. H. (1974). Environment and the structure of Organizations: A population ecology perspective. *Presented at Annual Meeting of American Sociology Association*, Montreal, Canada.
- Hardin, R. R. (1971). Collective action as an aggregate N-prisoner's dilemma. *Behavioral Sci.* 16, 472-481.
- Hardin, R. (1982). *Collective action*. Baltimore: John Hopkins University Press.

- Hargrave, T.J., & Van De Ven, A. H. (2006). A collective action model of institutional innovation. *Academy of Management Review*, 31, 864-888.
- Harhoff, D., Henkel, J., & von Hippel, E. (2003). Profiting from voluntary information spillovers: How users benefit by freely revealing their innovations. *Research Policy*, 32, 1753-1769.
- Heide, J. B. (1994). Interorganizational governance in marketing channels. *Journal of Marketing*, 58(1), 71-85.
- Hellmann, T., & Puri, M. (2000). The interaction between product market and financing strategy: the role of venture capital. *Review of Financial Studies*, 13, 959-84.
- Henkel, J. (2006). Selective revealing in open innovation processes: The case of embedded Linux. *Research Policy*, 35(7), 953-969.
- Hennart, J. (1988). A transaction costs theory of equity joint ventures. *Strategic Management Journal*, 9(4), 361-374.
- Hermann, S., G. Hertel, S. Niedner. (2000). *Linux study homepage*, Retrieved from [www.psychologie.uni-kiel.de/linux-study/writeup.html](http://www.psychologie.uni-kiel.de/linux-study/writeup.html).
- Hertel, G., Niedner, S., & Herrmann, S. (2003). Motivation of software developers in open source projects: An internet-based survey of contributors to the Linux kernel. *Research Policy*, 32, 1159-1177.
- Herzog, P., & Leker, J. (2007). Open vs. closed innovation strategies: Also different innovation cultures? *Proceedings of the R&D Management Conference (RADMA)*, Bremen, Germany, July 4-7, 2007.
- Heyman, J., & Ariely, D. (2004). Effort for payment- a tale of two markets. *Psychological Science*, 15(11), 787-793.
- Hill, C.W. L. (1995). National institutional structures, transaction cost economizing, and competitive advantage: The case of Japan. *Organization Science*, 6, 119-131.

- Hillenius, G. (2013). *City of Munich: "Migration to sustainable desktop completed successfully"*. Retrieved from European Commission-Joinup Information website:<https://joinup.ec.europa.eu/community/osor/news/city-munich-migration-sustainable-desktop-completed-successfully>
- Hillenius, G. (2016). *Open source initiative approves Quebec license*. Retrieved from European Commission-Joinup Information website:  
<https://joinup.ec.europa.eu/community/osor/news/open-source-initiative-approves-quebec-licence>
- Hillman, A. J., Withers, M. C, & Collins, B. J. (2009). Resource dependence theory: A review. *Journal of Management*, 35, 1404-1427.
- Himanen, P., Torvalds, L., & Castells, M. (2001). *The hacker ethic*. NY: Random House.
- Hirschheim, R., & Klein, H. K. (1989). Four paradigms of information systems development. *Communications of the ACM*, 32(10), 1199-1216.
- Holm, A. B., Günzel, F., & Ulhøi, J. P. (2013). Openness in innovation and business models: Lessons from the newspaper industry. *International Journal of Technology Management*, 61(3/4), 324-348.
- Holmstrom, B. (1999). Managerial incentive problems: A dynamic perspective. *Review of Economic Studies*, 66, 169-182.
- Holtgrewe, U., & Werle, R. (2001). De-Commodifying software? Open source software between business strategy and social movement. *Science Studies*, 14(2), 43-65.
- Howard, J., III, & Prividera, L. (2008). Freedom isn't free: A critical analysis of nationalism, militarism, and U.S. identity. *International and Intercultural Communication Annual*, 31, 151-174.
- Huston, L., & Sakkab, N. (2006). Connect and develop: Inside Procter & Gamble's new model for innovation. *Harvard Business Review*, 84, 58-66.

- Imperial, M. T. (2005). Using collaboration as a governance strategy: Lessons from six watershed management programs. *Administration & Society*, 37, 281-320.
- Inkpen, A. C., & Beamish, P. W. (1997). Knowledge, bargaining power, and the instability of international joint ventures. *Academy of Management Review*, 22, 177-202.
- Jensen, M. C., & Meckling, W. H. (1976). Theory of the firm: Managerial behavior, agency costs and ownership structure. *Journal of Financial Economics*, 3(4), 305-360.
- Jick, T. D. (1979). Mixing qualitative and quantitative methods: Triangulation in action. *Administrative Science Quarterly*, 24, 602-611.
- Johnson, M. W., & Suskewicz, J. (2009). How to jump-start the clean tech economy. *Harvard Business Review*, 87(11), 52-60.
- Johnson, R. B., Onwuegbuzie, A. J., & Turner, L. A. (2007). Toward a definition of mixed methods research. *Journal of Mixed Methods Research*, 1(2), 112-133.
- Kakalettris, G., Varoutas, D., Katsianis, D., Sphicopoulos, T., & Kouvas, G. (2004). Designing and implementing an open infrastructure for location-based, tourism-related content delivery. *Wireless Personal Communications*, 30(2-4), 153-165.
- Kale, P., Singh, H., & Perlmutter, H. (2000). Learning and protection of proprietary assets in strategic alliances: Building relational capital. *Strategic Management Journal*, 21, 217-237.
- Kale, P., & Singh, H. (2007). Building firm capabilities through learning: The role of the alliance learning process in alliance capability and firm-level alliance success. *Strategic Management Journal*, 28(10), 981-1000.
- Katz, M. L., & Shapiro, C. (1986). Technology adoption in the presence of network externalities. *Journal of Political Economy*, 94(4), 822-841.



- Katz, M. L., & Shapiro, C. (1985). Network externalities, competition and compatibility. *The American Economic Review*, 75(3), 424-440.
- Kelty, C. M. (2008). *Two bits: The cultural significance of free software*. Durham, NC: Duke University Press.
- Khanna, T., Gulati, R., & Nohria, N. (1998). The dynamics of learning alliances: Competition, cooperation, and relative scope. *Strategic Management Journal*, 19, 193-210.
- Khanna, T. (1996). Winner-take-all alliances. Harvard Business School Working Paper, No. 96-033.
- Kieser, A. (1997). Rhethoric and myth in management fashion. *Organization*, 4(1), 49-74.
- Kilduff, M., & Tsai, W. (2003). *Social networks and organizations*. London: SAGE.
- King, A. A., Tucci, C. L. (2002). Incumbent entry into new market niches: The role of experience and managerial choice in the creation of dynamic capabilities. *Management Science*, 48(2), 171-186.
- Kirschbaum, R. (2005). Open innovation in practice. *Research Technology Management*, 48(4), 24-28.
- Klandermans, B. (1997). *The social psychology of protest*. UK: Wiley.
- Klein, H. K., & Myers, M. D. (1999). A set of principles for conducting and evaluating interpretive field studies in information systems. *MIS Quarterly*, 23(1), 67-93.
- Kline, D. (2003). Sharing the corporate crown jewels. *MIT Sloan Management Review*, 44, 89-93.
- Kline, S. J. (1985). *Research, invention, innovation and production: Models and reality*, Rept.INN-1, Department of Mechanical Engineering, Stanford University.

- Kline, S. J., & Rosenberg, N. (1986). An overview of innovation. In R. Landau, & N. Rosenberg (Eds.), *The positive sum strategy: harnessing technology for economic growth* (pp.275-306). Washington, D.C.: National Academies Press.
- Knoke, D. (1990). *Political networks: The structural perspective*. New York: Cambridge University Press.
- Koenig, J. 2004. Seven open source business strategies for competitive advantage. *IT Manager's J.* (May 14).
- Kogut, B. (1988). Joint ventures: Theoretical and empirical perspectives. *Strategic Management Journal*, 9(4), 319-332.
- Kogut, B. (1989). The stability of joint ventures: Reciprocity and competitive rivalry. *Journal of Industrial Economics*, 38, 183-198.
- Kogut, B. (1991). Joint ventures and the option to expand and acquire. *Management Science*, 37(1), 19-33.
- Kogut, B., & Metiu, A. (2001). Open-Source software development and distributed innovation. *Oxford Review of Economic Policy*, 17(2), 248-264.
- Kogut, B., & Zander, U. (1992). Knowledge of the firm, combinative capabilities, and the replication of technology. *Organization Science*, 3(3), 383-397.
- Kohanski, D. (1998). *Moths in the machine*. New York: St. Martin's Griffin.
- Kolakowski, L. (1968). *The alienation of reason: A history of positivist thought*. New York: Doubleday.
- Kortum, S., & Lerner, J. (1999). Does venture capital spur innovation? *Working Paper No. 6846*, NBER.
- Kortum, S., & Lerner, J. (2000). Assessing the impact of venture capital on innovation. *Rand Journal of Economics*, 31, 674-92.

- Kumar, V., Gordon, B. R., & Srinivasan, K. (2011). Competitive Strategy for Open Source Software. *Marketing Science*, 30(6), 1066-1078.
- Kuzel, A. (1992). Sampling in qualitative inquiry. In B. F. Crabtree, & W. L. Miller (Eds.), *Doing qualitative research* (pp. 31-44). Thousand Oaks, CA: SAGE.
- Lakhani, K., & von Hippel, E. (2000). *How open source software works: "Free" user- to-user assistance*. Working Paper 4117, MIT Sloan School of Management, Cambridge, MA.
- Lakhani, K. R., & von Hippel, E. (2003). How open source software works: "Free" user-to-user assistance. *Research Policy*, 32(6), 923-943.
- Lakhani, K. R., & Wolf, R. G. (2005). Why hackers do what they do: Understanding motivation and effort in free/open source software projects. In J. Feller, B. Fitzgerald, S. Hissam, & K. Lakhani (Eds.), *Perspectives on free and open source software*. Cambridge: MIT Press.
- Lambert, D. M., & Cooper, M. C. (2000). Issues in supply chain management. *Industrial Marketing Management*, 29(1), 65-83.
- Lane, P. J., Koka, B. R., & Pathak, S. (2006). The reification of absorptive capacity: A critical review and rejuvenation of the construct'. *Academy of Management Review*, 31, 833-63.
- Langlois, R. N. (2003). The vanishing hand: The changing dynamics of industrial capitalism. *Industrial and Corporate Change*, 12(2), 351-385.
- Laursen, K., & Salter, A. J. (2006a). Open for innovation: The role of openness in explaining innovation performance among UK manufacturing firms. *Strategic Management Journal*, 27, 131-150.
- Laursen, K., & Salter, A. J. (2006b). *My precious technology: The role of legal appropriability strategy in shaping innovative performance*. Working Paper. Tanaka Business School, Imperial College London.

- Lavie, D. (2006). The competitive advantage of interconnected firms: An extension of the resource-based view. *The Academy of Management Review*, 31(3), 638-658.
- Lawrence, B. W. (1997). The black box of organizational demography. *Organization Science*, 8, 1-22.
- Lazzarotti, V., Manzini, R., & Pellegrini, L. (2011). Firm-specific factors and the openness degree: A survey of Italian firms. *European Journal of Innovation Management*, 14(4), 412 – 434.
- Leana C, Van Buren HJ III (1999) Organizational social capital and employment practices. *Academy of Management Review*, 24(3), 538–555.
- Lee, A. S., & Baskerville, R. L. (2003). Generalizing generalizability in information systems research. *Information Systems Research*, 14(3), 221-243.
- Lee, S-Y T., Kim, H-W, Gupta, S. (2009). Measuring open source software success. *Omega*, 37, 426-438.
- Lepak, D. P., Smith, K. G., Taylor, S. (2007). Value creation and value capture: A multilevel perspective. *Academy of Management Review*, 32(1), 180–194.
- Leifer, R., & Delbecq, A. (1978). Organizational/environmental interchange: A model of boundary spanning activity. *Academy of Management Review*, 40-50.
- Lerner J, & Tirole, J. (2001). The open source movement: Key research questions. *European Economic Review*, 45(4), 819–826.
- Lerner, J., Tirole. J. (2002). Some simple economics of open source. *Journal of Industrial Economics*, 50(2), 197-234.
- Lerner, J., Tirole. J. (2005). The scope of open source licensing. *Journal of Law, Economics, and Organization*, 21(1), 20–56.

- Lerner, J., Pathak, P. A., & Tirole, J. (2006). The dynamics of open-source contributors. *The American Economic Review*, 96(2), 114-118.
- Levin, R. C., Klevorick, A. K., Nelson, R., & Winter, S. (1987). Appropriating the returns from industrial research and development. *Brookings Papers on Economic Activity*, 3, 783-831.
- Levine, S., & White, P. E. (1961). Exchange as a conceptual framework for the study of interorganizational relationships. *Administrative Science Quarterly*, 5(4), 583-601.
- Levinthal, D. A., & March, J. G. (1993). The myopia of learning. *Strategic Management Journal*, 14, 94-112.
- Li, Y., Tan, C.-H., Xu, H., & Teo, H.-H. (2011). Open source software adoption: Motivations of adopters and amotivations of non-adopters. *The DATA BASE for Advances in Information Systems*, 42(2), 76-94.
- Lichtenthaler, U. (2008). Open innovation in practice: An analysis of strategic approaches to technology transactions. *IEEE Transactions on Engineering Management*, 55(1), 148-157.
- Lichtenthaler, U. (2011). Open innovation: past research, current debates, and future directions. *Academy of Management Perspectives*, 25(1), 75-93.
- Lichtenthaler, U., & Lichtenthaler, E. (2009). A capability-based framework for open innovation: complementing absorptive capacity. *Journal of Management Studies*, 46(8), 1315-1338.
- Lincoln, Y. S., & Guba, E. G. (1985). *Naturalistic inquiry*. Beverly Hills, CA: SAGE.
- Lindsey, L. (2008). Blurring firm boundaries: The role of venture capital in strategic alliances. *The Journal of Finance*, LXIII(3), 1137-1168.
- Linux Foundation. (2014). *Collaborative development trends report*. Retrieved from <https://www.linux.com/publications/collaborative-development-trends-report-2014>

- Linux Foundation. (2016). *Moving toward professionalization: Rising need for open source skills in 2016*. Retrieved from <http://go.linuxfoundation.org/download-2016-open-source-jobs-report>
- Locke, E.A. (2007). The case for inductive theory building. *Journal of Management*, 33(6), 867-890.
- Locke, K. (1996). Rewriting the discovery of grounded theory after 25 years? *Journal of Management Inquiry*, 5(3), 239-245.
- Lofland, J., & Lofland, L. H. (1995). *Analyzing social setting: A guide to qualitative observation and analysis* (2<sup>nd</sup> ed.). Belmont, CA: Wadsworth.
- Lorenzoni, G., & Lipparini, A. (1999). The leveraging of interfirm relationships as a distinctive organizational capability: A longitudinal study. *Strategic Management Journal*, 20, 317-38.
- Luhmann, N. (1995). *Social systems*. Stanford, CA: Stanford University Press.
- Lundell, B., & van der Linden, F. (2013). Open source software as open innovation: Experiences from the medical domain. In Jenny S. Z. Eriksson Lundström, Mikael Wiberg, Stefan Hrastinski, Mats Edenius, Pär J. Ågerfalk (Eds.), *Managing open innovation technologies* (pp. 3-16). Berlin: Springer Berlin Heidelberg.
- Luo, C.-M., & Chang, H.-F. (2011). SME competitive strategy: Learning from Taiwan's ODM industry. *Business Strategy Series*, 12(3), 107-114.
- Macaulay, S. (1963). Non-contractual relations in business: A preliminary study. *American Sociological Review*, 28, 55-70.
- Macneil, I. R. (1974). The many futures of contracts. *Southern California Law Review*, 47, 691-816.
- Macneil, I. R. (1980). *The new social contract*. New Haven, CT.: Yale University Press.

- Magretta, J. (2002). Why business models matter. *Harvard Business Review*, 80(5), 86-92.
- Mansfield, G. M., & Fourie, L. C. H. (2004). Strategy and business models—strange bedfellows? A case for convergence and its evolution into strategic architecture. *South African Journal of Business Management*, 35(1), 35-44.
- Mansfield, E., & Wagner, S. (1975). Organizational and strategic factors associated with probabilities of success in industrial R&D. *The Journal of Business*, 48(2), 179-198.
- March, J. G. (1991). Exploration and exploitation in organizational learning. *Organization Science*, 2, 71-87.
- March, J. G., Sproull, L. S., & Tamuz, M. (1991). Learning from samples of one or fewer. *Organization Science*, 2, 1-13.
- Marshall, C., & Rossman, G. B. (1989). *Designing qualitative research*. California: Sage Publications.
- March, J. G., & Olsen, J. P. (1975). The uncertainty of the past: Organizational learning under ambiguity. *European Journal of Political Research*, 3, 147-171.
- March, J. G., & Simon, H. A. (1958). *Organizations*. New York: Wiley.
- Marshall, C., & Rossman, G. B. (1995). *Designing qualitative research*. London: Sage Publications.
- Mariti, P., & Smiley, R. H. (1983). Co-operative agreements and the organization of industry. *Journal of Industrial Economics*, 31(4), 437-451.
- Mayo, M.C., & Brown, G.S. (1999). Building a competitive business model. *Ivey Business Journal*, 63(3), 18-23.
- Maxwell, E. (2006). Open standards, open source, and open innovation: Harnessing the benefits of openness. *Innovations: Technology, Governance, Globalization*, 1(3), 119-176.

- McGregor, S. L. T., & Murnane, J. A. (2010). Paradigm, methodology and method: Intellectual integrity in consumer scholarship. *International Journal of Consumer Studies*, 34(4), 419-427.
- Méndez-Durón, R., & García, C.E. (2009). Returns from social capital in open source software networks. *Journal of Evolutionary Economics*, 19, 277-295.
- Merton, R. K., Fiske, M., & Kendall, P. L. (1990). *The focused interview: A manual of problems and procedures* (2<sup>nd</sup> ed.). New York: Free Press.
- Miller, D., & Shamsie, J. (1996). The resource-based view of the firm in two environments: The Hollywood film studios from 1936 to 1965. *Academy of Management Journal*, 39, 519-543.
- Miles, M., & Huberman, A. (1994). *Qualitative data analysis: An expanded source book*. Thousand Oaks, CA: Sage.
- Mina, A., Bascavusoglu-Moreau, E., & Hughes, A. (2014). Open service innovation and the firm's search for external knowledge. *Research Policy*, 43, 853-866.
- Mizruchi, M. (1983). Who controls whom? An examination of the relation between management and boards of directors in large corporations. *Academy of Management Review*, 8, 426-35.
- Mockus, A., Fielding, R. T., & Herbsleb, J. (2002). Two case studies of open source software development: Apache and Mozilla. *ACM Trans. Software Engrg. Methodology*, 11(3), 309-346.
- Moffett, J. (1968). *Teaching the Universe of Discourse*. Boston: Houghton Mifflin.
- Mohr, J., & Spekman, R. (1994). Characteristics of partnership success. *Strategic Management Journal*, 15(2), 135-152.
- Morgan, L., & Finnegan, P. (2010). Open innovation in secondary software firms: An exploration of managers' perceptions of open source software. *The DATA BASE for Advances in Information Systems*, 41(1), 76-95.



- Morgan, L., & Finnegan, P. (2014). Beyond free software: An exploration of the business value of strategic open source. *Journal of Strategic Information Systems*, 23, 226-238.
- Morris, M., Schindehutte, M., & Allen, J. (2005). The entrepreneur's business model: Toward a unified perspective. *Journal of Business Research*, 58(6), 726-735.
- Morse, J.M., Stern, P.N., Corbin, J., Bowers, B., Charmaz, K., & Clarke, A.E. (2009). *Developing grounded theory: The second generation*. Walnut Creek, CA: Left Coast Press.
- Mowery, D. C. (1983). The relationship between intrafirm and contractual forms of industrial research in American manufacturing, 1900-1940. *Explorations in Economic History*, 20, 351-374.
- Mowery, D. C. (1988). *International collaborative ventures in U.S. manufacturing* (Ed.). Cambridge, MA: Ballinger.
- Mowery, D. C. (2009). Plus ca change: Industrial R&D in the "third industrial revolution". *Industrial and Corporate Change*, 18(1), 1-50.
- Mowery, D. C., Oxely, J. E., & Silvermann, B. S. (1996). Strategic alliances and interfirm knowledge transfer. *Strategic Management Journal*, 17, 77-92.
- Myers, M. D. (2013). *Qualitative research in business & management* (2<sup>nd</sup> ed.). London: SAGE.
- Nachmias, D., & Nachmias, C. (1992). *Research methods in the social sciences*. NY: St. Martin's.
- Nakamura, M., Shaver, J. M., & Yeung, B. (1996). An empirical investigation of joint venture dynamics: Evidence from US-Japan joint ventures. *International Journal of Industrial Organization*, 14, 521-541.
- Nambisan, S., & Wilemon, D. (2000). Software development and new product development: Potential for cross-domain knowledge sharing. *IEEE Transactions on Engineering Management*, 47(2), 211-220.

- Nahapiet, J., & Ghoshal, S. (1998). Social capital, intellectual capital, and the organizational advantage. *Academy of Management Review*, 23(2), 242-266.
- Nelson, R., & Winter, S. G. (1982). The Schumpeterian tradeoff. *American Economic Review*, 72(1), 114-33.
- Nelson, R. R., & Winter, S. (1982). *An evolutionary theory of economic change*. Cambridge, MA: Harvard University Press, Balknap Press.
- Nickerson, J. A., Silverman, B. S., Zenger, T. R. (2007). The ‘problem’ of creating and capturing value. *Strategic Organisation*, 5(3), 211–225.
- Niosi, J. (1995). *Flexible innovation – technological alliances in Canadian industry*. Canada, Montreal: McGill-Queen’s University Press.
- Niosi, J. (1999). Fourth-generation R&D: From linear models to flexible innovation. *Journal of Business Research*, 45, 111–117.
- Niosi, J., & Godin, B. (1999). Canadian R&D abroad management practices. *Research Policy*, 28, 215-230.
- Niosi, J., & Hade, N. (1995). Biotechnology. In J. Niosi, *Flexible innovation-Technological alliances in Canadian industry* (pp. 77-98). Montreal: McGill-Queen’s University Press.
- Niosi, J., & Rivard, J. (1990). Canadian technology transfer to developing countries through small and medium-sized enterprises. *World Development*, 18(11), 1529-1542.
- Nonneke, B., Preece, J. (2000). Lurker demographics: Counting the silent. *Proceedings of CHI 2000* (pp. 73-80). The Hague: Association for Computing Machinery, ACM Press.
- Normandin, P-A., (2014, July 2). Ménage technologique à la Ville de Montréal. *LaPress*. Retrieved from <http://www.lapresse.ca/actualites/montreal/201407/02/01-4780369-menage-technologique-a-la-ville-de-montreal.php>.

- North, D. C. (1990). *Institutions, institutional change and economic performance*. Cambridge, England: Cambridge University Press.
- OECD. (2008). Globalization and open innovation. OECD, Paris.
- O'Mahony, S. (2003). Guarding the commons: How community managed software projects protect their work. *Research Policy*, 32(7), 1179–1198.
- O'Mahony, S., & Ferraro, F. (2007). The emergence of governance in an open source community. *Academy of Management*, 50(5), 1079–1106.
- Oliver, C. (1990). Determinants of interorganizational relationships: Integration and future directions. *Academy of Management Review*, 15, 241–265.
- Olson, M. (1967). *The logic of collective action*. Cambridge, MA: Harvard University Press.
- Osterwalder, A., & Pigneur, Y. (2002). An ebusiness model ontology for modeling ebusiness. *BLED 2002 Proceedings* (Paper 2). Available at: <http://aisel.aisnet.org/bled2002/2>.
- Osterwalder, A., & Pigneur, Y. (2010). *Business model generation: A handbook for visionaries, game changers, and challengers*. Wiley.
- O'Toole, L. J. (1997). Treating networks seriously: Practical and research-based agendas in public administration. *Public Administration Review*, 57, 45–52.
- Oxley, J. E. (1997). Appropriability hazards and governance in strategic alliances: A transaction cost approach. *Journal of Law Economics and Organization*, 13, 389–407.
- Ozman, M. (2009). Inter-firm networks and innovation: A survey of the literature. *Economics of Innovation and New Technology*, 18(1), 39–67.
- Paré, A., & Smart, G. (1994). Observing genres in action: Towards a research methodology. In A. Freedman & P. Medway (Eds.), *Genre and the New Rhetoric* (pp. 146–154). London: Taylor & Francis.

- Parker, G. G., van Alstyne, & Choudary, S. P. (2016). *Platform Revolution*. New York: W. W. Norton & Company.
- Parkhe, A., Wasserman, S., & Ralston, D. A. (2006). New frontiers in network theory development. *The Academy of Management Review*, 31(3), 560-568.
- Pavlicek, R. C. (2000). *Embracing insanity: Open source software development*. Indianapolis IN: Sams.
- Penrose, E. T. (1959). *The theory of the growth of the firm*. New York: Wiley.
- Perens, B. (1998). The open source definition. <http://perens.com/Articles/OSD.html>
- Perr, J., Appleyard, M. M. & Sullivan, P. (2010). Open for business: Emerging business models in open source software. *International Journal of Technology Management*, 52(3/4), 432-456.
- Pfeffer, J. (1987). A resource dependence perspective on interorganizational relations. In M. S. Mizruchi, & M. Schwartz (Eds.), *Interorganizational relations: The structural analysis of business* (pp.22-55). Cambridge, UK: Cambridge University Press.
- Pfeffer, J., & Nowak, P. (1976). Joint ventures and interorganizational interdependence. *Administrative Science Quarterly*, 21, 398-418.
- Peteraf, M. A. (1993). The cornerstones of competitive advantage: A resource-based view. *Strategic Management Journal*, 14, 179-191.
- Pfeffer, J. (1972). Merger as a response to organizational interdependence. *Administrative Science Quarterly*, 17, 382-94.
- Pfeffer, J., & Salancik, G. R. (1978). *The external control of organizations: A resource dependence perspective*. New York: Harper & Row.
- Pfeffer, J., & Salancik, G. R. (2003). *The external control of organizations: A resource dependence perspective*: Stanford, CA: Stanford University Press.

- Piller, F., Vossen, A., & Ihl, C. (2012). From social media to social product development: The impact of social media on co-creation of innovation. *Die Unternehmung*, 65(1), 7-27.
- Piller, F., & West, J. (2014). Firms, users, and innovation an interactive model of coupled open innovation. In Henry W. Chesbrough, Wim Vanhaverbeke, Joel West (Eds.), *New frontiers in open innovation* (pp.29-49). Oxford: Oxford University Press.
- Pisano, G. P. (1990). The R&D boundaries of the firm: An empirical analysis. *Administrative Science Quarterly*, 35(1), 153-176.
- Pisano, G. P. (1989). Using equity participation to support exchange: Evidence from the biotechnology industry. *Journal of Law, Economics, and Organization*, 5, 109-126.
- Pisano, G. P., Russo, M. V., & Teece, D. (1988). Joint ventures and collaborative agreements in the telecommunications equipment industry'. In D. Mowery (ed.), *International Collaborative Ventures in US Manufacturing*. Ballinger, Cambridge, MA, pp. 23-70.
- Polanyi, M. (1967). *The tacit dimension*. New York: Doubleday.
- Popper, K. (1983). *Realism and the aim of science: From the postscript to the logic of scientific discovery*. Totowa, NJ: Rowman and Littlefield.
- Porter, M. E. (1980a). *Competitive strategy*. New York: Free Press.
- Porter, M. E. (1980b). Industry structure and competitive strategy: Keys to profitability. *Financial Analysts Journal*, 36(4), 30-41.
- Porter, M. E., & Stern, S. (2001). Innovation: Location matters. *Sloan Management Review*, 42(4), 28-43.
- Porter, M. E., & Fuller, M. B. (1986). Coalitions and global strategy. In M. E. Porter (Ed.), *Competition in Global Industries* (pp. 315-344). Boston, MA: Harvard Business School Press.

- Powell, W. (1987). Hybrid organizational arrangements: New form or transitional development. *California Management Review*, 30, 67–87.
- Powell, W. W. (1990). Neither market nor hierarchy: network forms of organization. In B. M. Staw, & L. L. Cummings (Eds.), *Research in organizational behavior* (vol. 12, pp. 295- 336). Greenwich, CT: JAI Press.
- Powell, W. W., Koput, K. W., & Smith-Doerr, L. (1996). Interorganizational collaboration and the locus of innovation: Networks of learning in biotechnology. *Administrative Science Quarterly*, 41, 116-145.
- Prahalad C. K., & Hamel, G. (1990). The core competence of the corporation. *Harvard Business Review*, 68(3), 79–91.
- Prencipe, A., Davies, A., & Hobday, M. (Eds.). (2003). *The business of systems integration*. Oxford: Oxford University Press.
- Priem, R. L. (2007). A consumer perspective on value creation. *The Academy of Management Review*, 32(1), 219-235.
- Priem, R. L., & Butler, J.E. (2001). Is the resource-based “view” a useful perspective for strategic management research? *Academy of Management*, 26(1), 22-40.
- Provan, K. G., & Kenis, P. (2008). Modes of network governance: Structure, management, and effectiveness. *Journal of Public Administration Research and Theory: J-PART*, 18(2), 229-252.
- Purdy, M., Robinson, M. C., & Wei, K. (2012). Three new business models for “the open firm”. *Strategy & Leadership*, 40(6), 36–41.
- Rai, A., Lang, S. S., & Welker, R. B. (2002). Assessing the validity of IS success models: An empirical test and theoretical analysis. *Information Systems Research*, 13(1), 50-69.
- Raymond, E. (1999). *The cathedral and the bazaar: Musings on Linux and open source by an accidental revolutionary*. Sebastopol, CA: O'Reilly & Associates.

- Rayna, T., & Striukova, L. (2010). Large-scale open innovation: Open source vs. patent Pools. *International Journal of Technology Management*, 52(3/4), 477-496.
- Reed, R., & DeFillippi, R. J. (1990). Causal ambiguity, barriers to imitation, and sustainable competitive advantage. *Academy of Management Review*, 15, 88-102.
- Richardson, G. B. (1972). The organization of industry. *Economic Journal*, 82, 883-896.
- Riehle, D. (2007). The economic motivation of open source software: Stakeholder perspectives. *Computer*, 40(4), 25-32.
- Roberts, J.A., Hann, I-H., & Slaughter, S. A. (2006). Understanding the motivations, participation, and performance of open source software developers: A longitudinal study of the apache projects. *Management Science*, 52(7), 984-999.
- Robbins, L. (1932). *An essay on the nature and significance of economic science*. New York: New York University Press.
- Robins, J. (2013). Editorial. *Long Range planning*, 46, 417-418.
- Roijakkers, N., & Hagedoorn, J. (2006). Inter-firm R&D partnering in pharmaceutical biotechnology since 1975: Trends, patterns, and networks. *Research Policy*, 35, 431-446.
- Romero, D., & Molina, A. (2011). Collaborative networked organisations and customer communities: Value co-creation and co-innovation in the networking era. *Production Planning & Control*, 22(5/6), 447-472.
- Ronde, T. (1999). *Trade secrets and information sharing*. Unpublished working paper, University of Mannheim.
- Rosenberg, N. (1982). *Inside the black box: technology and economics*. New York: Cambridge University Press.

- Rosenberg, N. (1990). Why do firms do basic research (with their own money)? *Research Policy*, 19, 165–174.
- Rothaermel, F. T. (2001). Incumbent's advantage through exploiting complementary assets via interfirm cooperation. *Strategic Management Journal*, 22, 687–99.
- Rothwell, R., Freeman, C., Horseley, A., Jervis, V.T.P., & Townsend, J. (1974). Sappho Updated—Project Sappho Phase II. *Research Policy*, 3, 204–225.
- Rothwell, R. (1994). Towards the fifth-generation innovation process. *International Marketing Review*, 11(1), 7–31.
- Rothwell, R., & Zegveld, W. (1985). *Reindustrialization and technology*. USA: Longman, Harlow.
- Rumelt, R. P. (1984). Towards a strategic theory of the firm. In R. B. Lamb (Ed.), *Competitive strategic management* (pp. 556–570). Englewood Cliffs, NJ: Prentice-Hall.
- Savoir-faire Linux, Inc. (2017). *Tackling three FFmpeg's technological challenges to further shape TR-03 standards for broadcasters*. Retrieved from <https://blog.savoirfairelinux.com/en-ca/p/savoir-faire-linux/>
- Sako, M. (1991). The role of “trust” in Japanese buyer-supplier relationships. *Ricerche Economiche*, XLV, 449–474.
- Sandulli, F. D., & Chesbrough, H. W. (2009a). Open business models: las dos caras de los modelos de negocio abiertos. *Universia Business Review*, 22, 12–39.
- Sandulli, F. D., & Chesbrough, H. W. (2009b). The two faces of open business models. Retrieved from SSRN: <http://ssrn.com/abstract=1325682> or <http://dx.doi.org/10.2139/ssrn.1325682>.
- Scacchi, W. (2002). Understanding the requirements for developing open source software systems. *IEE Software Proc.* 149(1), 24–39.



- Schaarschmidt, M., Walsh, G., & von Kortzfleisch, H. F.O. (2015). How do firms influence open source software communities? A framework and empirical analysis of different governance modes. *Information and Organization*, 25, 99-114.
- Schensul, S., Schensul, J. J., & LeCompte, M. D. (1999). *Essential ethnographic methods: Observations, interviews, and questionnaires. Volume 2 of the ethnographer's toolkit*. Walnut Creek, California: AltaMira Press.
- Schumpeter, J. A. (1934). *The theory of economic development*. Cambridge, MA: Harvard University Press.
- Seddon, P. B. (1997). A respecification and extension of the DeLone and McLean model of IS success. *Information Systems Research*, 8(3), 240-253.
- Seddon, P. B., Staples, S., Patnayakuni, R., & Bowtell, M. (1999). Dimensions of information systems success. *Communications of the Association for Information Systems*, 2(20), 1-61.
- Seelos, C., & Mair, J. (2007). Profitable business models and market creation in the context of deep poverty: A strategic view. *Academy of Management Perspectives*, 21, 49-63.
- Sen, R., Singh, S. S., & Borle, S. (2012). Open source software success: Measures and analysis. *Decision Support Systems*, 52, 364-372.
- Shah, S. K. (2006). Motivation, governance, and the viability of hybrid forms in open source software development. *Management Science*, 52(7), 1000-1014.
- Smith, P., Cavalcante, S., Kesting, P., & Ulhøi, J. (2010). Opening up the business model: A multi-dimensional view of firms' inter-organizational innovation activities. In *Proceedings of the 11th International CI-Net Conference* (pp. 1-13). Zürich, CH.
- Sorenson, O., & Rogan, M. (2014). (When) do organizations have social capital? *Annual Review of Sociology*, 40, 261-280.

- Spender, J.-C. (1994). Organizational knowledge, collective practice, and Penrose rents. *International Business Review*, 3(4), 353-367.
- Spender, J.-C. (1992). Limits to learning from the west. *The International Executive*, 34(September/October), 389-410.
- Stake, R.E. (1995). *The art of case study research*. Thousand Oaks, CA: SAGE.
- Stam, W. (2009). When does community participation enhance the performance of open source software companies? *Research Policy*, 38, 1288-1299.
- Stearns, T. M., Hoffman, A. N., & Heide, J. B. (1987). Performance of commercial television stations as an outcome of interorganizational linkages and environmental conditions. *Academy of Management Journal*, 30, 71-90.
- Stewart, D. W., & Zhao, Q. (2000). Internet marketing, business models, and public policy. *Journal of Public Policy & Marketing*, 19(2), 287- 96.
- Stewart, K.J. & Gosain, S. (2006). The impact of ideology on effectiveness in open source software development team. *MIS Quarterly*, 30(2), 291-314.
- Stewart, K. J., Ammeter, A. P., & Maruping, L. M. (2006). Impact of license choice and organizational sponsorship on success in open source software development projects. *Information System Research*, 17(2), 126-144.
- Stiglitz, J. (1987). Learning to learn: localized learning and technological progress. In P. Dasgupta & P. Stoneman (Eds.), *Economic policy and technological performance* (pp. 125-153). Cambridge: Cambridge University Press.
- Strauss, A. L., & Corbin, J. (1990). *Basics of qualitative research: Grounded theory procedures and techniques*. Newbury Park, CA: SAGE.
- Subramaniam, C., Sen, R., Nelson, M. L. (2009). Determinants of open source software project success: A longitudinal study. *Decision Support Systems*, 46, 576-585.

- Sturgeon, T. J. (2002). Modular production networks: A new American model of industrial organization. *Industrial and Corporate Change*, 11(3), 451–496.
- Sutton, R. I., & Staw, B. M. (1995). What theory is not. *Administrative Science Quarterly*, 40(3), 371–384.
- Swanson, G. E. (1992). Doing this together—Some basic forms of agency and structure in collective action and some explanations. *Soc. Psych. Quart.*, 55(2), 94-117.
- Swanson, E. B. (1994). Information systems innovation among organizations. *Management Science*, 40(9), 1069–1092.
- Tapscott, D. (2000). *Digital capital: Harnessing the power of business webs*. Boston: Harvard Business School Press.
- Taran, Y., Boer, H., & Lindgren, P. (2015). A business model innovation typology. *Decision Sciences*, 46(2), 301-331.
- Tashakkori, A., & Teddlie, C. (1998). *Mixed methodology: Combining qualitative and quantitative approaches*. Thousand Oaks, CA: Sage.
- Tashakkori, A., & Teddlie, C. (2003). *Handbook of mixed methods in social and behavioral research*. Thousand Oaks, CA: Sage.
- Teece, D. J. (1977). Technology transfer by multinational firms. *Economic Journal*, 87, 242-261.
- Teece, D. J. (1992). Competition, cooperation, and innovation- organizational arrangements for regimes of rapid technological progress. *Journal of Economic Behavior and Organization*, 18, 1-25.
- Teece, D. J. (2006). Reflections on “profiting from innovation. *Research Policy*, 35, 1131–1146.

- Teece, D. J. (2007). Explicating dynamic capabilities: The nature and microfoundations of (sustainable) enterprise performance. *Strategic Management Journal*, 28(13), 1319–1350.
- Teece, D. J. (2010). Business models, business strategy and innovation. *Long Range Planning*, 43, 172-194.
- Teece, D. J., & Pisano, G. (1994). The dynamic capabilities of firms: an introduction. *Industrial and Corporate Change*, 3, 537-556.
- Teece, D. J., Pisano, G., & Shuen, A. (1997). Dynamic capabilities and strategic management. *Strategic Management Journal*, 18(7), 509-533.
- Thelen, M. H., Dollinger, S. J., & Kirkland, K. D. (1979). Imitation and response uncertainty. *Journal of Genetic Psychology*, 135, 139-52.
- Therrien, Y. (2013, September 21). Logiciel libre: Québec aurait pu économiser 265 millions\$. *LeSoleil*. Retrieved from <http://www.lapresse.ca/le-soleil/affaires/techno/201309/20/01-4691638-logiciel-libre-quebec-aurait-pu-economiser-265-millions-.php>.
- Thompson, J. (1962). Organizations and output transactions. *American Journal of Sociology*, 68, 309- 325.
- Timmers, P. (1998). Business models for electronic markets. *Electronic Markets*, 8(2), 3-8.
- Torvalds, L., & Diamond, D. (2001). *Just for Fun: the Story of an Accidental Revolutionary*. New York: Harper Business.
- Tracy, S. J. (2013). *Qualitative research methods*. Malaysia: Wiley-Blackwell.
- Tripsas, M. (1997). Unraveling the process of creative destruction: Complementary assets and incumbent survival in the typesetter industry. *Strategic Management Journal*, 18, 119-142.

- Trott, P., & Hartmann, D. (2009). Why 'open innovation' is old wine in new bottles. *International Journal of Innovation Management*, 13(4), 715–736.
- Tuomi, I. (2002), *Networks of innovation. Change and meaning in the age of the Internet*. Oxford: Oxford University Press.
- Ulhøi, J. P. (2004). Open source development: A hybrid in innovation and management theory. *Management Decision*, 42(9), 1095–1114.
- Uzzi, B. (1997). Social structure and competition in interfirm networks: the paradox of embeddedness. *Administrative Science Quarterly*, 42, 35–67.
- Van de Ven, A. H. (1976). On the nature, formation, and maintenance of relations among organizations. *Academy of Management Review*, 1(4), 24–36.
- Van de Ven, A. H. (1989). Nothing is quite so practical as a good theory. *Academy of Management Review*, 14(4), 486–489.
- Van de Ven, A. H., & Walker, G. (1984). The dynamics of interorganizational coordination. *Administrative Science Quarterly*, 29, 598–621.
- Van de Vrande, V., Vanhaverbeke, W., & Gassmann, O. (2010). Broadening the scope of open innovation: past research, current state and future directions. *International Journal of Technology Management*, 52(3/4), 221–235.
- Van de Vrande, de Jong, J. P. J., Vanhaverbeke, W., & Rochemont, M.D. (2009). Open innovation in SMEs: Trends, motives and management challenges. *Technovation*, 29, 423–437.
- Vanhaverbeke, W., & Cloudt, M. (2006). Open innovation in value constellations. In: Henry Chesbrough, Wim Vanhaverbeke and Joel West (Eds.), *Open innovation: researching a new paradigm* (pp.258–281). USA: OxfordUniversity Press.
- Vanhaverbeke, W., & Cloudt, M. (2014). Theories of the firm and open innovation. In Henry W. Chesbrough, Wim Vanhaverbeke, Joel West (Eds.), *New frontiers in open innovation* (pp. 256–278). Oxford: Oxford University Press.

- Vanhaverbeke, W., van de Vrande, V., & Chesbrough, H. (2008). Understanding the advantages of open innovation practices in corporate venturing in terms of real options. *Creativity and Innovation Management*, 17(4), 251-258.
- Varian, H. (2002). If there was a new economy, why wasn't there a new economics? *New York Times*, January 17, C2.
- Vernon, R. (1983). Organizational and institutional responses to international risk. In R. Herring (Ed.), *Managing International Risk*. New York: Cambridge University Press.
- Vickers, J. (1985). Pre-emptive patenting, joint ventures, and the persistence of oligopoly. *International Journal of Industrial Organization*, 3, 261-273.
- Vogt, W. P. (2007). *Quantitative research methods for professionals*. USA: Pearsons Education, Inc.
- Von Hippel, E. (1988). *The sources of innovation*. New York: Oxford University Press.
- Von Hippel, E. (1994). Sticky information and the locus of problem solving: implications for innovation. *Management Science*, 40(4), 429-439.
- Von Hippel, E. (2001). User toolkits for innovation. *Journal of Product Innovation Management*, 18(4), 247-257.
- Von Hippel, E., & Katz, R. (2002). Shifting innovation to users via toolkits. *Management Science*, 48(7), 821-834.
- Von Hippel, E. (2005). *Democratizing innovation*. Cambridge, MA: MIT Press.
- Von Hippel, E. (2010). Open user innovation. In B. H. Hall, & N. Rosenberg (Eds.), *Handbook of the economics of innovation* (pp. 411-427). The Netherlands: North-Holland.
- Von Hippel, E., & de Jong, Jeroen P.J. (2010). Open, distributed and user-centered: Towards a paradigm shift in innovation policy. *EIM Research Report*,

Number H201009, URL: <http://www.entrepreneurship-sme.eu/pdf-ez/H201009.pdf>.

- Von Hippel, E., & von Krogh, G. (2003). Open source software and the 'private-collective' innovation model: Issues for organization science, *Organization Science*, 14(2), 209-223.
- Von Hippel, E., & von Krogh, G. (2006). Free revealing and the private- collective model for innovation incentives. *R&D Management*, 36(3), 295-306.
- Von Hippel, E., Susumo, Ogawa, & de Jong, Jeroen P. J. (2012). The age of consumer innovator. *Sloan Management Review*, 53(1), 27-33.
- Von Krogh, G. (2003). Open source software development. *MIT Sloan Management Review*, 44(3), 14-18.
- Von Krogh, G., Spaeth, S., & Lakhani, K. R. (2003). Community, joining, and specialization in open source software innovation: A case study. *Research Policy*, 32(7), 1217-1241.
- Von Krogh, G., & Spaeth, S. (2007). The open source software phenomenon: Characteristics that promote research. *Strategic Information Systems*, 16(3), 236-253.
- Von Krogh, G., & von Hippel, E. (2003). Special issue on open source software development. *Research Policy*, 32, 1149-1157.
- Von Krogh, G., & von Hippel, E. (2006). The promise of research on open source software. *Management Science*, 52(7), 975-983.
- Walker, G., Kogut, B., & Shan, W. (1997). Social capital, structural holes and the formation of an industry. *Organizational Science*, 8(2), 109-125.
- Walsham, G. (1993). *Interpreting information systems in organizations*. Chichester, UK: Wiley.
- Walsham, G. (1995a). Interpretive case studies in is research: nature and method. *European Journal of Information Systems*, 4(2), 74-81.

- Walsham, G. (1995b). The emergence of interpretivism in IS research. *Information Systems Research*, 6(4), 376-394.
- Walsham, G. (1997). Actor-network theory and IS research: Current status and future prospects. In Lee, A. S., Liebenau, J., & DeGross, J. L. (Eds.), *Information systems and qualitative research* (pp.466-480). London, UK: Chapman and Hall.
- Wang, Y., & Zhou, Z. (2012). Can open innovation approach be applied by latecomer firms in emerging countries? *Journal of Knowledge-based Innovation in China*, 4(3), 163-173.
- Wang, L., Jaring, P., & Arto, W. (2009). Developing a conceptual framework for business model innovation in the context of open innovation. In *Proceedings of the 3<sup>rd</sup> IEEE International Conference on Digital Ecosystems and Technologies* (pp. 460-465). IEEE.
- Weber, S. (2004). *The success of open source*. Cambridge, Mass.: Harvard University Press.
- Weiblen, T., Frankenberger, K., & Gassmann, O. (2013). The open business model: Towards a common understanding of an emerging concept. *European Academy of Management*, 26-29 June 2013.
- Welch, Ivo. (1992). Sequential sales, learning and cascades. *Journal of Finance*, 47(2), 695-732.
- Wernerfelt, B. (1984). A resource-based view of the firm. *Strategic Management Journal*, 5(2), 171-180.
- West, J. (2003). How open is open enough? Melding proprietary and open source platform strategies. *Research Policy*, 32(7), 1259-1285.
- West, J. (2006). Does appropriability enable or retard open innovation? In Chesbrough, Henry, Wim Vanhaverbeke, and Joel West (Eds.), *Open innovation: Researching a new paradigm* (pp.109-133). Oxford: Oxford University Press.



- West, J. (2007). Value capture and value networks in open source vendor strategies. In *Proceedings of the 40th Annual Hawaii International Conference on System Sciences* (HICSS'07).
- West, J., & Bogers, M. (2014). Leveraging external sources of innovation: A review of research on open innovation. *Journal of Product Innovation Management*, 31(4), 814-831.
- West, J., & Dedrick, J. (2001). Open source standardization: the rise of Linux in the network era, knowledge. *Technology and Policy*, 14(2), 88-112.
- West, J., & Gallagher, S. (2006). Challenges of open innovation: The paradox of firm investment in open-source software, *R&D Management*, 36(3), 319-331.
- West, J., & Lakhani, K. (2008). Getting clear about communities in open innovation. *Industry and Innovation*, 15(2), pp. 223-261.
- West, J., & O'Mahony, S. (2005). Contrasting community building in sponsored and community founded open source projects. In *Proceedings of the 38th Annual Hawai'i International Conference on System Sciences* (IEEE), Waikoloa, Hawaii.
- West, J., & O'Mahony, S. (2008). The role of participation architecture in growing sponsored open source communities. *Industry and Innovation*, 15(2), 145-168.
- West, J., Vanhaverbeke, W., & Chesbrough, H. (2006). Open innovation: A research agenda. In Henry Chesbrough, Wim Vanhaverbeke, and Joel West (Eds.), *Open Innovation: Researching a New Paradigm*. Oxford: Oxford University Press, pp. 285-307.
- Westphal, J. D., & Zajac, E. J. (1995). Who shall govern? CEO/board power, demographic similarity, and new director selection. *Administrative Science Quarterly*, 40, 60-83.
- Whetten, D. A. (1989). What constitutes a theoretical contribution? *Academy of Management Review*, 14(4), 490-495.

- Wichmann, T. (2002). Use of open source software in firms and public institutions. Evidence from Germany, Sweden and UK. Free/Libre and open source software: Survey and study, FLOSS final report, International Institute of Infonomics, Berlecom Research GmbH.
- Williamson, O. E. (1975). *Markets and hierarchies: Analysis and antitrust implications*. New York: Basic Books.
- Williamson, O. E. (1979). Transaction-cost economics: the governance of contractual relations. *Journal of Law and Economics*, 22(2), 233–261.
- Williamson, O. E. (1981). The economics of organization: the transaction cost approach. *American Journal of Sociology*, 87, 548–577.
- Williamson, O. E. (1985). *The Economic Institutions of Capitalism: Firms, markets, relational contracting*. New York: Free Press.
- Williamson, O. E. (1991). Comparative economic organization: the analysis of discrete structural alternative. *Administrative Science Quarterly*, 36(2), 269–296.
- Williamson, O. E. (1998). Transaction cost economics: how it works; where it is headed. *De economist*, 146(1), 23–58.
- Williamson, O.E. (2005). *Chapter 3: Transaction cost economics*. Handbook of Industrial Organization, 1, 135–182.
- Williamson, O. E. (2010). Transaction costs economics: The natural progression. *American Economic Review*, 100, 673–690.
- Willis, J.W. (2007). *Foundations of Qualitative Research-Interpretive and Critical Approaches*. SAGE Publications, Inc.: Thousand Oaks, California, USA.
- Woods, D., & Guliani, G. (2005). *Open source for the enterprise*. Sebastopol, CA: O'Reilly Media.

- Yan, A., & Gray, B. (1994). Bargaining power, management control, and performance in united states-china joint ventures: A comparative case study. *Academy of Management Journal*, 37, 1478-1517.
- Yan, A., & Gray, B. (2001). Antecedents and effects of parent control in international joint ventures. *Journal of Management Studies*, 38, 393-416.
- Yin, R. K. (1984). *Case study research: Design and methods*. Thousand Oaks, CA: SAGE.
- Yin, R. K. (1994). *Case study research: Design and methods* (2nd ed.). Thousand Oaks, CA: SAGE.
- Yin, R. K. (2003). *Applications of case study research* (2nd ed.). Thousand Oaks, CA: SAGE.
- Yin, R. K. (2009). *Case study research* (4th ed.). USA: SAGE.
- Yin, R. K. (2011). *Qualitative research from start to finish*. New York: The Guilford Press.
- Zahra, S. A., & Newey, L. (2009). Maximizing the impact of organization science: Theory-building at the intersection of disciplines and/or fields. *Journal of Management Studies*, 46, 1059-1075.
- Zajac, E. J., & Olsen, C. P. (1993). From transaction cost to transactional value analysis: Implications for the study of interorganizational strategies. *Journal of Management Studies*, 30(1), 131-145.
- Zeitlyn, D. (2003). Gift economies in the development of open source software: Anthropological reflections. *Research Policy*, 32(7), 1287-1291.
- Zivney, T.L., & Bertin, W.J. (1992). Publish or perish: What the competition is Really doing. *The Journal of Finance*, 47(1), 295-329.
- Zott, C., & Amit, R. (2007). Business model design and the performance of entrepreneurial firms. *Organization Science*, 18, 181-199.

- Zott, C., & Amit, R. (2008). The fit between product market strategy and business model: Implications for firm performance. *Strategic Management Journal*, 29, 1-26.
- Zott, C., & Amit, R. (2009). The business model as the engine of network-based strategies. In P. R. Kleindorfer & Y. J. Wind (Eds.), *The network challenge* (pp. 259-275). Upper Saddle River, NJ: Wharton School Publishing.
- Zott, C., & Amit, R. (2010). Designing your future business model: an activity system perspective. *Long Range Planning*, 43, 216-226.
- Zott, C., Amit, R., & Massa, L. (2011). The business model: recent developments and future research. *Journal of Management*, 37(4), 1019-1042.